

الجمهورية العربية السورية  
المعهد العالي للعلوم التطبيقية والتكنولوجيا  
قسم المعلومات  
العام الدراسي 2023/2024

مشروع سنة رابعة

# e-commerce recommendation system

إعداد

غفار محمد حيدر

إشراف

د. مصطفى الدقاق

م. عمار مخلوف

28/8/2024



## الإهداء

إلى سر وجودي في هذا الكون، إلى من عطر لي عبق الحياة، إلى من أفتخر بكوني قطعة منها

أمي الحبيبة

إلى الحارس الذي أطاب لي غمض العين، إلى المنارة التي تضيء دربي

أبي الغالي

إلى عدتي في شدي، إلى الذين كانوا سنداً اتكأت عليهم في لحظات ضعفي، إلى من كان نجاحي بسببهم ولأجلهم

إخوتي الأعزاء

إلى أخي وصديقي وشريكي الغائب الحاضر دوماً في قلوبنا

خضر خليل

إلى من أبعدته الغربة وقربته ال rocket league

ملهم قطان

إلى كل من وقف بجانب يوماً

إلى جميع أصدقائي وزملائي الذين شاركوني مقاعد الدراسة منذ بداية طريق العلم

ليس كل ما يلمع ذهباً

by William Shakespeare

## كلمة شكر

أتقدم بجزيل الشكر إلى كل من وقف بجاني، وساهم في نجاح هذا العمل .  
أخص بالشكر الدكتور المشرف مصطفى الدقاق والمهندس عمار مخلوف  
لتوجيهاتهم ونصائحهم طيلة فترة المشروع.

أخيراً، أشكر من كان سنداً ودعماً لي في كل اللحظات ... أهلي  
وأصدقائي.



الملخص

**Abstract**

## Table of Contents

1-	الفصل الأول	1
1.1-	تمهيد	1
2.1-	هدف المشروع	1
2-	الفصل الثاني	2
1.2-	الذكاء الاصطناعي ARTIFICIAL INTELLIGENCE	2
1.1.2-	تعلم الآلة Machine Learning	2
2.1.2-	التعلم العميق Deep Learning	4
2.2-	الشبكات العصبونية الصناعية (NN) NEURAL NETWORKS	5
1.2.2-	الشبكات العصبونية كاملة الارتباط Fully Connected Neural Networks	6
2.2.2-	الشبكات العصبونية التكرارية (RNN)	7
3.2.2-	ذاكرة طويلة قصيرة المدى (LSTM)	7
3.2-	عملية التعلم LEARNING في الشبكات العصبونية	8
1.3.2-	توابع التنشيط (Activation Functions)	8
2.3.2-	توابع الخسارة Loss Functions	11
-3	الفصل الثالث	14
1.3-	استخدام التعلم العميق في فهم سلوك المستخدم	14
2.3-	أنظمة التوصية الأكثر شيوعاً في الوقت الحاضر	15
3.3-	نماذج متقدمة في أنظمة التوصية	15
-4	الفصل الرابع	18
-5	الفصل الخامس	20
1.5-	الهيكل العام للمشروع	20
2.5-	إعداد بيئة العمل	21
3.5-	إنشاء الإضافة (EXTENSION) الأساسية	21
4.5-	البحث عن نماذج التوصية	24
5.5-	اختيار MICROSOFT RECOMMENDERS	26
6.5-	إعداد وتجهيز مجموعة البيانات (DATASET)	28
1.6.5-	تنسيق البيانات المدخلة (Input Data Format)	28
2.6.5-	معالجة البيانات (data preprocessing)	28
3.6.5-	إنشاء بيانات وهمية للتدريب عليها	30
7.5-	تدريب النموذج	32
8.5-	دمج النموذج مع OPENCART باستخدام MLFLOW	36
1.8.5-	محاولة استخدام Flask API	36
2.8.5-	الانتقال إلى MLflow	36





3	الشكل 1 الفرق بين خوارزميات تعلم الآلة والخوارزميات التقليدية .....
5	الشكل 2 الفرق بين تعلم الآلة والتعلم العميق .....
6	الشكل 3 الفرق بين تعلم الآلة والتعلم العميق .....
8	الشكل 4 بنية شبكة LSTM .....
10	الشكل 5 الفرق بين تابعي التنشيط sigmoid و ReLU .....
11	الشكل 6 تابع التنشيط Leaky ReLU .....
20	الشكل 7 الهيكل العام للمشروع .....
21	الشكل 8 جدول تفاعل المستخدمين مع المنتجات .....
22	الشكل 9 MVCL design pattern .....
22	الشكل 10 هرمية الملفات في قسم المدير .....
23	الشكل 11 هرمية الملفات في قسم المستخدم .....
23	الشكل 12 كيفية ظهور الاقتراحات للمستخدم .....
28	الشكل 13 مثال عن البيانات المدخلة .....
34	الشكل 14 تغير AUC مع التكرارات .....
34	الشكل 15 تغير قيم Logloss مع التكرارات .....
35	الشكل 16 تغير قيم Mean MRR مع التكرارات .....
35	الشكل 17 تغير قيم Group AUC مع التكرارات .....

الجدول 1 مقارنة الأداء بحسب AUC	17
الجدول 2 مقارنة الأداء بحسب F1-score	17
الجدول 3 مقارنة بين المكاتب المتاحة	25

# 1-الفصل الأول

## مقدمة عن المشروع

يتضمن هذا الفصل التعريف بالمشروع ومتطلباته.

### 1.1- تمهيد

يشهد العالم مؤخراً تطوراً ملحوظاً في تقنيات الذكاء الاصطناعي، حيث انطلقت مجالات جديدة ومبشرة تحقق مزيجاً مذهلاً من الإبداع البشري والتقنيات الحديثة. من هذه المجالات الرائجة التي تثير الدهشة هو فهم سلوك البشر، حيث تعتمد هذه التقنية على تحليل بيانات المستخدمين وسلوكياتهم أثناء تصفحهم عبر الإنترنت. تساهم هذه التقنية في إثراء تجارب المستخدمين من خلال تقديم توصيات مخصصة ودقيقة تتوافق مع تفضيلاتهم واحتياجاتهم، وتحسين استراتيجيات التسويق عن طريق فهم أعمق لرغبات العملاء أثناء تصفحهم للمنتجات أو شرائهم لها عبر الإنترنت، فضلاً عن تعزيز ولاء العملاء ورفع مستوى التفاعل مع المنتجات. يلقي هذا العمل نظرة على أحد أهم الأوراق البحثية التي تسلط الضوء على كيفية عمل هذه التقنيات والأسس العلمية وراءها، كما يقدم نبذة عن التحديات التي تعترض العمل في هذا المجال، مثل معالجة الكم الهائل من البيانات وضمان خصوصية المستخدم وتقديم توصيات تتسم بالدقة والابتكار، والكثير من التحديات الأخرى التي سيجري تناولها لاحقاً.

### 2.1- هدف المشروع

يهدف هذا المشروع إلى تطوير إضافة (Extension) لنظام إدارة محتوى التسوق الإلكتروني OpenCart، تعتمد على تحليل سلوك المستخدمين لاستنتاج واقتراح المنتجات التي تتناسب مع اهتماماتهم وتفضيلاتهم. يتيح هذا النظام المتقدم للتجار تقديم تجربة تسوق مخصصة لكل مستخدم، مما يزيد من فرص تحويل الزوار إلى مشتريين ويعزز من ولاء العملاء، وذلك من خلال تقديم توصيات دقيقة وشخصية تعتمد على الأنماط السلوكية والتفاعل مع المنتجات.

## 2-الفصل الثاني

# الدراسة النظرية

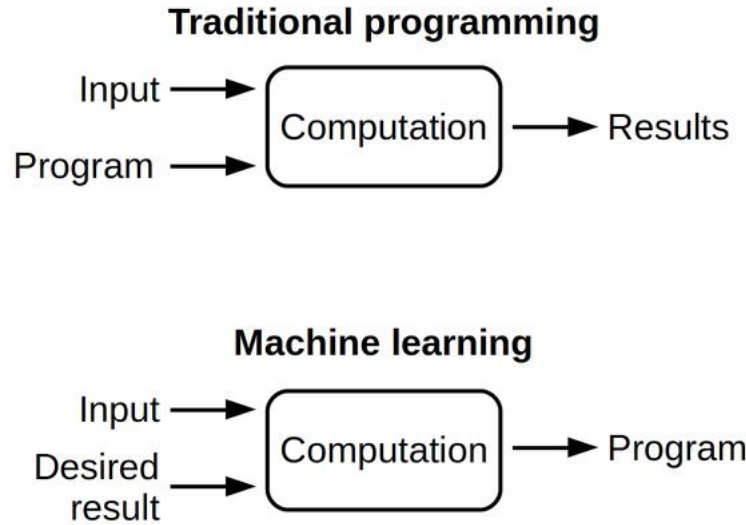
نستعرض في هذا الفصل مفهوم الذكاء الصناعي، مروراً بمفهومَي تعلم الآلة والتعلم العميق، يلي ذلك شرح نظري عن الشبكات العصبونية الصناعية.

### 1.2- الذكاء الصناعي Artificial Intelligence

الذكاء الاصطناعي (Artificial Intelligence) هو أحد مجالات علوم الحاسوب الذي يهدف إلى تطوير أنظمة وبرمجيات قادرة على محاكاة الذكاء البشري وأداء المهام التي تتطلب تفكيراً معقداً وذكياً. يركز هذا المجال على الاستفادة من القدرات الذهنية للإنسان وتطبيقها في الأنظمة الحاسوبية. يتمحور الذكاء الاصطناعي حول بناء أنظمة تكنولوجية قادرة على تحليل البيانات والمعلومات بطرق مشابهة للطريقة التي يفكر بها الإنسان، واتخاذ قرارات مبنية على تلك التحليلات. يعتمد الذكاء الاصطناعي بشكل كبير على مفهوم تعلم الآلة (Machine Learning)، وهو أحد فروع الذكاء الاصطناعي الذي يمكن الأنظمة من تعديل سلوكها وأدائها بناءً على البيانات التي تتعامل معها، بدلاً من الاعتماد كلياً على البرمجة التقليدية. في الوقت الحالي، يعد الذكاء الاصطناعي من التقنيات المتقدمة والحديثة التي تؤثر بشكل كبير على العديد من المجالات والصناعات مثل التجارة الإلكترونية، الرعاية الصحية، التصنيع، النقل، الموارد البشرية، الألعاب، والتسويق الرقمي، حيث تُستخدم تطبيقات الذكاء الاصطناعي لتحسين الكفاءة واتخاذ قرارات أكثر دقة وتسهيل المهام المعقدة التي قد تكون صعبة على البشر. ومع ذلك، ورغم التقدم العلمي والتقني الكبير، فإن هذا المجال لا يزال يواجه العديد من التحديات والعوائق، من أبرزها التحديات التقنية المتعلقة بقوة المعالجة، والتخزين، والتحسين المستمر للأداء.

### 1.1.2- تعلم الآلة Machine Learning

تعلم الآلة هو أحد فروع الذكاء الاصطناعي الذي يركز على تطوير نماذج ونظم قادرة على تعلم الأنماط والقواعد من البيانات، واستخدام هذا التعلم لاتخاذ قرارات أو التنبؤ بالنتائج. يُستخدم تعلم الآلة في مجموعة واسعة من التطبيقات، مثل تصفية البريد الإلكتروني المزعج أو الضار، وتقديم توصيات الأفلام على منصات الترفيه، والتعرف على الصوت والنص، والقيادة الذاتية للمركبات، وغيرها من المجالات. يوضح الشكل 1 الفرق بين خوارزميات تعلم الآلة والخوارزميات التقليدية.



الشكل 1 الفرق بين خوارزميات تعلم الآلة والخوارزميات التقليدية

### أقسام تعلم الآلة

يتكون مجال تعلم الآلة من أربعة أقسام رئيسية:

- **التعلم الخاضع للإشراف (Supervised Learning)** في هذا القسم، يتم تدريب النماذج باستخدام أزواج من البيانات تتكون من مدخلات ومخرجات متوقعة. الهدف من النموذج هو التعرف على الأنماط والروابط بين المدخلات والمخرجات لتقديم تنبؤات دقيقة عند مواجهة مدخلات جديدة.
- **التعلم غير الخاضع للإشراف (Unsupervised Learning)** بعكس التعلم الخاضع للإشراف، يُدرب النموذج على بيانات بدون مخرجات متوقعة. يهدف النموذج إلى اكتشاف الأنماط والترتيبات المخفية داخل بيانات المدخلات. تُستخدم خوارزميات مثل التجميع (لجمع نقاط البيانات المتشابهة معاً) وتقليل الأبعاد (لضغط البيانات عالية الأبعاد إلى تمثيل منخفض الأبعاد) بشكل شائع في هذا النوع من التعلم، وغالباً ما يُستخدم للكشف عن الهيكل الأساسي للبيانات.
- **التعلم شبه الخاضع للإشراف (Semi-supervised Learning)** يجمع هذا النوع بين التعلم الخاضع للإشراف والتعلم غير الخاضع للإشراف. في هذا النوع، تُستخدم مجموعة بيانات تحتوي على مدخلات مع مخرجات مصنفة ومجموعة بيانات أخرى غير مصنفة. الهدف هو تحسين أداء النموذج باستخدام البيانات المصنفة والبيانات غير المصنفة معاً. إحدى الطرق الشائعة هي تدريب النموذج على البيانات المصنفة ثم استخدام البيانات غير المصنفة لتحسين

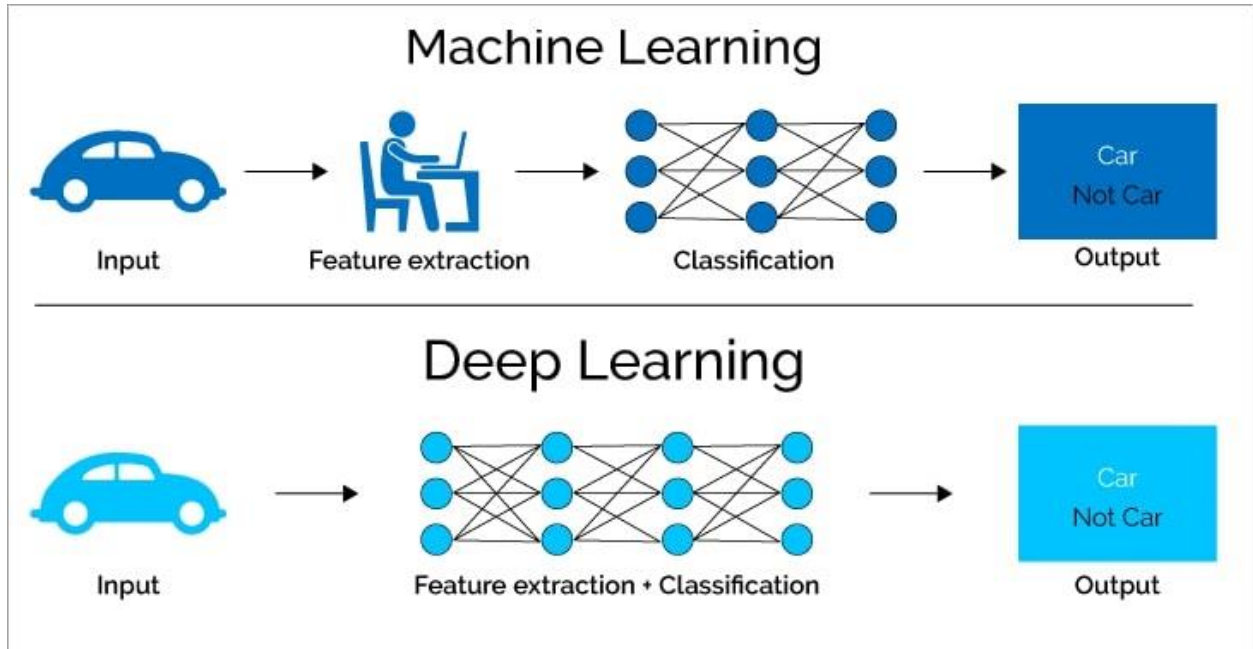
النموذج عبر تعديل الأوزان. يُستخدم التعلم شبه الخاضع للإشراف بشكل خاص عندما تكون البيانات المصنفة قليلة ولكن البيانات غير المصنفة متوفرة بكميات كبيرة، ويُطبق في مجالات مثل تصنيف النصوص والتعرف على الكلام وتحسينات في تطبيقات الرؤية الحاسوبية.

- **التعلم المعزز: (Reinforcement Learning)** في هذا النوع، يُدرب النموذج من خلال نظام مكافآت وعقوبات. يتعلم النموذج من خلال التفاعل مع البيئة واتخاذ القرارات التي تزيد من المكافآت وتقلل من العقوبات. الهدف هو تحسين استراتيجية النموذج بناءً على التجارب السابقة والمكافآت أو العقوبات التي تلقاها. يستخدم التعلم المعزز في تطبيقات متنوعة، مثل الألعاب والروبوتات وأنظمة التحكم الذاتي.

تغطي هذه الأقسام الأساليب المختلفة لتعلم الآلة، كل منها يتميز بتطبيقاته ومزاياه الفريدة، مما يسمح للنماذج بتحسين قدراتها في معالجة وتحليل البيانات المعقدة واتخاذ قرارات مبنية على البيانات.

## 2.1.2- التعلم العميق Deep Learning

التعلم العميق هو أحد تقنيات تعلم الآلة التي تعتمد على الشبكات العصبونية الصناعية متعددة الطبقات. تُستوحى هذه الشبكات من هيكل الشبكة العصبية في الدماغ البشري، وتتكون من طبقات متعاقبة من العقد أو الوحدات الحسابية التي ترتبط بالمدخلات والمخرجات. يتم تدريب الشبكات العصبية العميقة باستخدام كميات كبيرة من البيانات، مما يسمح لها باكتشاف الأنماط والسياقات في البيانات بشكل تلقائي دون الحاجة إلى تحديد الميزات بشكل يدوي. تُظهر نماذج التعلم العميق كفاءة عالية في التعرف على الأنماط المعقدة في الصور والنصوص والأصوات والبيانات الأخرى، مما يمكنها من تقديم رؤى وتنبؤات دقيقة. تُستخدم تقنيات التعلم العميق في أتمتة مجموعة من المهام التي تتطلب عادةً ذكاءً بشرياً، مثل تصنيف الصور وتحليل النصوص، والتعرف على الصوت. يوضح الشكل 2 الفرق بين تعلم الآلة والتعلم العميق.



الشكل 2 الفرق بين تعلم الآلة والتعلم العميق

## 2.2- الشبكات العصبونية الصناعية (NN) Neural Networks

الشبكات العصبونية الاصطناعية هي نماذج حاسوبية تستلهم بنيتها من الجهاز العصبي البيولوجي. تتكون الشبكة العصبونية الاصطناعية من وحدات حسابية تُعرف بالعصبونات (Perceptions)، ويتم تنظيم هذه الوحدات في طبقات متعددة ومتزايدة. عادةً، تتألف الشبكة العصبونية من ثلاثة أنواع من الطبقات:

1. **طبقة الإدخال (Input Layer):** هي الطبقة الأولى في الشبكة التي تستقبل البيانات المدخلة، ثم تقوم بتمريرها إلى الطبقات التالية لمعالجتها.

2. **الطبقات المخفية (Hidden Layers):** تقع بين طبقة الإدخال وطبقة الإخراج، وفيها يتم تنفيذ عمليات التعلم واستخراج الخصائص والميزات من البيانات. تعتبر هذه الطبقات هي جوهر عملية التعلم في الشبكة العصبونية.

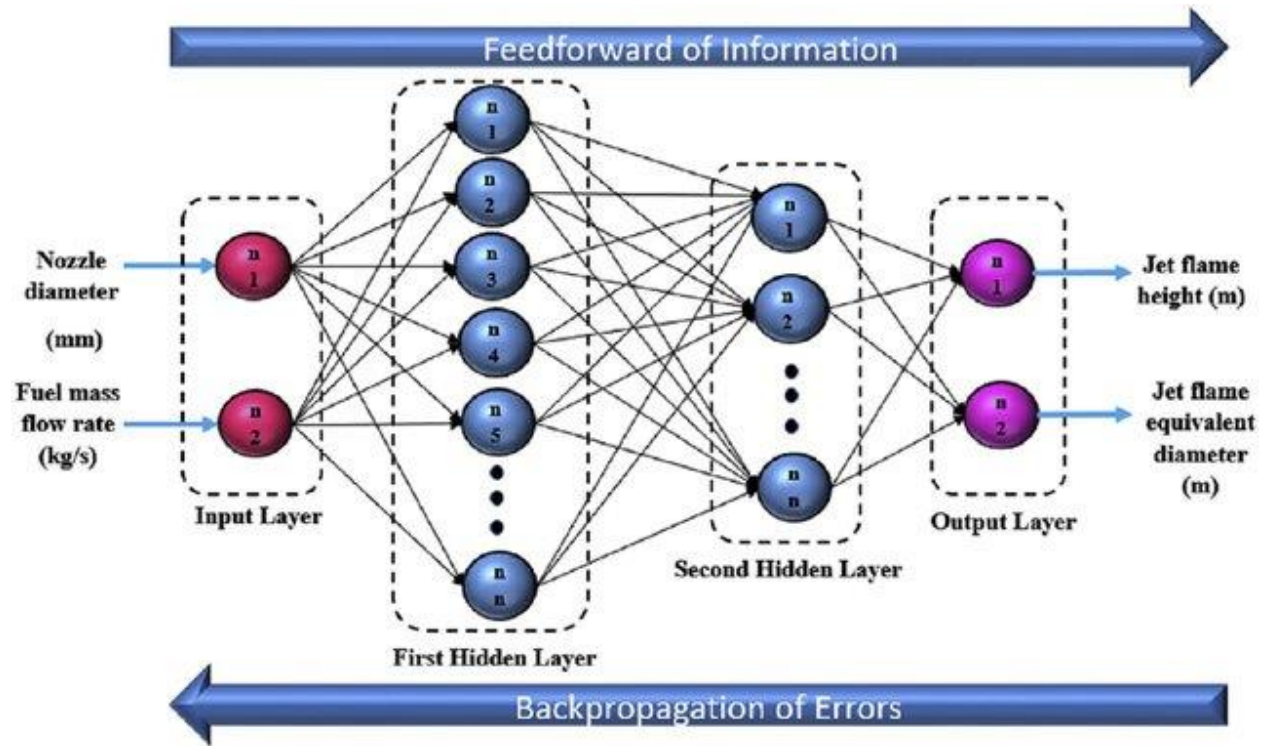
3. **طبقة الإخراج (Output Layer):** هي الطبقة الأخيرة في الشبكة، والتي تقوم بإنتاج النتائج أو التصنيفات المتوقعة بناءً على البيانات المدخلة والمعالجة التي تمت في الطبقات المخفية.

تتصل المدخلات بالعصبونات في الطبقة الأولى من خلال وصلات (Connections) لكل منها وزن محدد (Weight)، كما تتصل العصبونات فيما بينها عبر وصلات ذات أوزان موزونة. يتم حساب الخرج عن طريق عملية تُسمى الانتشار الأمامي (Forward Propagation)، حيث يكون دخل كل عصبون عبارة عن مجموع مخرجات العصبونات التي تُغذيها، مضروباً في الأوزان المحددة لتلك الوصلات، بالإضافة إلى معامل ثابت يُسمى الانحياز (Bias). بعد ذلك، يخضع الدخل للمعالجة من قبل



العصبون باستخدام تابع تنشيط (Activation Function)، الذي سنناقش آلية اختياره لاحقاً، ثم يتم تمرير الخرج إلى العصبون التالي، وهكذا حتى الوصول إلى طبقة الإخراج.

يتم التعلم في الشبكة العصبية عن طريق تعديل أوزان الوصلات بين العصبونات استناداً إلى بيانات التدريب (Training Data)، والتي تتضمن أزواجاً من المدخلات والمخرجات التي يجب على الشبكة تعلمها. توفر بيانات التدريب تغذية راجعة (Back Propagation) حول دقة الأوزان التي تم تعيينها، بناءً على مدى تطابق مخرجات الشبكة المتوقعة مع القيم الحقيقية للمخرجات المقابلة لنفس المدخلات في بيانات التدريب. يوضح الشكل 3 الفرق بين تعلم الآلة والتعلم العميق.



الشكل 3 الفرق بين تعلم الآلة والتعلم العميق

## 1.2.2- الشبكات العصبونية كاملة الارتباط Fully Connected Neural Networks

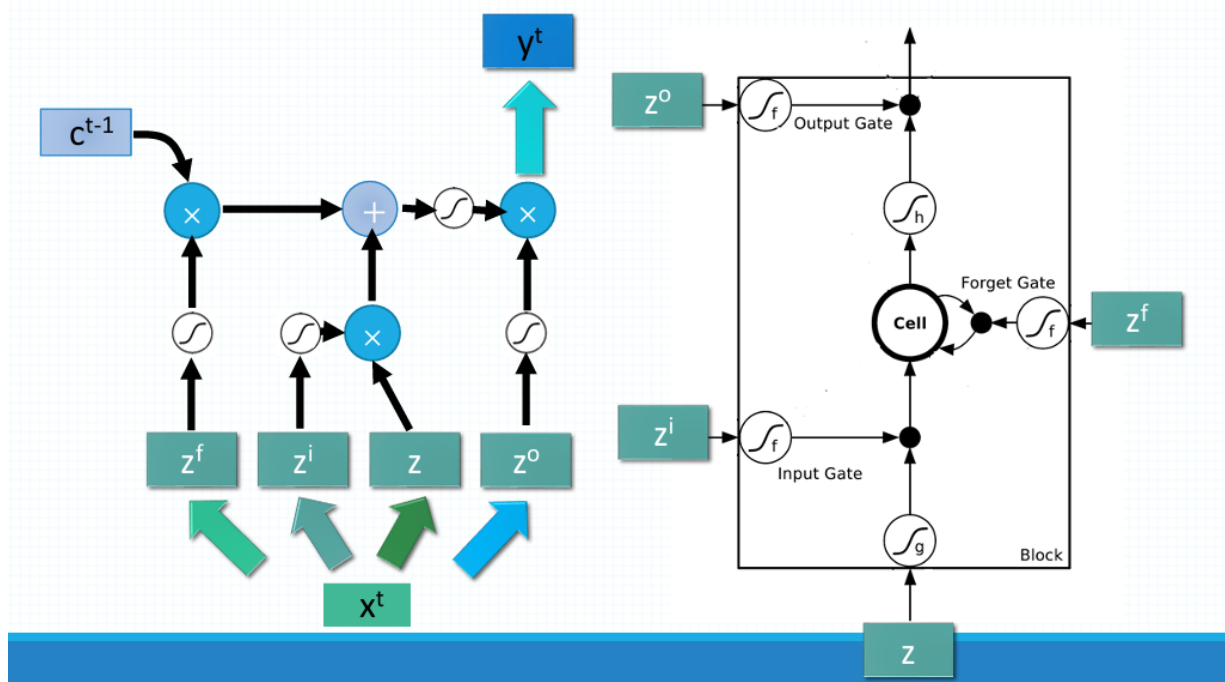
تُعرف أيضاً باسم الشبكات العصبية الكثيفة (Dense Neural Networks)، وتُعد من أبسط وأكثر النماذج شيوعاً في مجال تعلم الآلة. في هذا النوع من الشبكات، يرتبط كل عصبون في طبقة معينة بجميع العصبونات في الطبقة السابقة، مما يخلق شبكة كاملة الارتباط. تُعتبر الشبكات العصبية الكاملة الارتباط قوية وقادرة على التعامل مع مجموعة متنوعة من المشاكل والتطبيقات، ولكنها تتطلب عدداً كبيراً من الوسيطات (parameters) مع زيادة عمق الشبكة، مما يؤدي إلى زيادة كبيرة في الحمل الحسابي. لهذا السبب، تُستخدم هذه الشبكات بشكل شائع في التطبيقات التي تتطلب معالجة بيانات ذات أبعاد منخفضة، مثل مسائل التصنيف البسيطة، حيث تكون المتطلبات الحسابية أقل تعقيداً.

## 2.2.2- الشبكات العصبية التكرارية (RNN)

الشبكات العصبية التكرارية (Recurrent Neural Networks - RNN) هي نوع من الشبكات العصبية الاصطناعية مصممة للتعامل مع البيانات المتسلسلة والزمنية. تتميز RNN بقدرتها على تذكر المعلومات السابقة في سلسلة البيانات بفضل الوصلات التكرارية، مما يجعلها فعالة جداً في معالجة البيانات التي تتطلب النظر في السياق السابق، مثل النصوص، والكلام، والموسيقى، والسلاسل الزمنية في الأسواق المالية. تعتمد الشبكات العصبية التكرارية على فكرة تمرير المعلومات من خطوة زمنية إلى الخطوة التالية داخل الشبكة، مما يسمح للنموذج بالتعلم من الأنماط الزمنية وتعقيدها. ومع ذلك، فإن الشبكات العصبية التكرارية التقليدية تواجه تحديات في معالجة التسلسلات الطويلة بسبب مشاكل مثل تلاشي أو انفجار التدرج (Vanishing or Exploding Gradient Problem)، مما يؤدي إلى صعوبة تعلم الأنماط التي تعتمد على معلومات بعيدة في الزمن.

## 3.2.2- ذاكرة طويلة قصيرة المدى (LSTM)

ذاكرة طويلة قصيرة المدى (Long Short-Term Memory - LSTM) هي نوع متقدم من الشبكات العصبية التكرارية (RNN) تم تصميمه للتغلب على مشاكل التلاشي والانفجار في التدرج التي تواجهها الشبكات التكرارية التقليدية عند التعامل مع تسلسلات طويلة من البيانات. تم تصميم وحدات LSTM بطريقة تسمح لها بتذكر المعلومات لفترات زمنية طويلة عن طريق استخدام ثلاثة مكونات رئيسية: بوابة الإدخال (Input Gate)، وبوابة النسيان (Forget Gate)، وبوابة الإخراج (Output Gate). تعمل بوابة الإدخال على تحديد المعلومات الجديدة التي يجب إضافتها إلى حالة الذاكرة، بينما تتحكم بوابة النسيان في المعلومات التي يجب التخلص منها أو الاحتفاظ بها، وأخيراً تقوم بوابة الإخراج بتحديد المعلومات التي سيتم استخدامها لإنتاج الخرج في الخطوة الزمنية الحالية. بفضل هذا الهيكل المعقد، تتمتع LSTM بقدرة أكبر على التعلم من التسلسلات الطويلة والمعقدة، مما يجعلها مناسبة للتطبيقات التي تتطلب فهماً عميقاً للسياق، مثل الترجمة الآلية، والتعرف على الكلام، والتنبؤ بالنصوص. يوضح الشكل 4 بنية شبكة LSTM.



الشكل 4 بنية شبكة LSTM.

### 3.2- عملية التعلم Learning في الشبكات العصبونية

تتم عملية التعلم في الشبكات العصبونية من خلال عدة خطوات متتالية تشمل معالجة البيانات المدخلة، وتقييم أداء النموذج، وتحديث الأوزان. أولاً، يتم معالجة المعطيات المدخلة باستخدام توابع التنشيط (Activation Functions) التي تعمل على تحويل الإشارات الواردة إلى مخرجات يمكن معالجتها بشكل فعال داخل الشبكة. بعد ذلك، يتم تقييم أداء النموذج باستخدام تابع الخسارة (Loss Function) الذي يقيس الفروقات بين المخرجات الفعلية والمخرجات المتوقعة. بناءً على هذا التقييم، يتم ضبط الأوزان باستخدام خوارزمية تعلم مثل خوارزمية الانتشار العكسي (Backpropagation) لتقليل الخطأ وتحسين أداء النموذج.

#### 1.3.2- توابع التنشيط (Activation Functions)

توابع التنشيط هي المكونات الأساسية للشبكات العصبونية، حيث تقوم بتحديد ما إذا كان عصبون معين سيتم تفعيله أم لا بناءً على الإشارة المدخلة. تلعب هذه التوابع دوراً حيوياً في تحسين العلاقات بين البيانات المدخلة والمخرجات المرغوبة، وتساهم بشكل مباشر في أداء الشبكة العصبونية. لذا، فإن اختيار تابع التنشيط المناسب هو خطوة مهمة لتحقيق نتائج أفضل في عملية التدريب. فيما يلي نناقش بعض أشهر توابع التنشيط غير الخطية المستخدمة في التعلم العميق.

### 1.1.3.2 - تابع سيجمويد (Sigmoid Function)

تابع سيجمويد هو أحد توابع التنشيط الشائعة في الشبكات العصبية. يُعرف هذا التابع بالعلاقة التالية:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

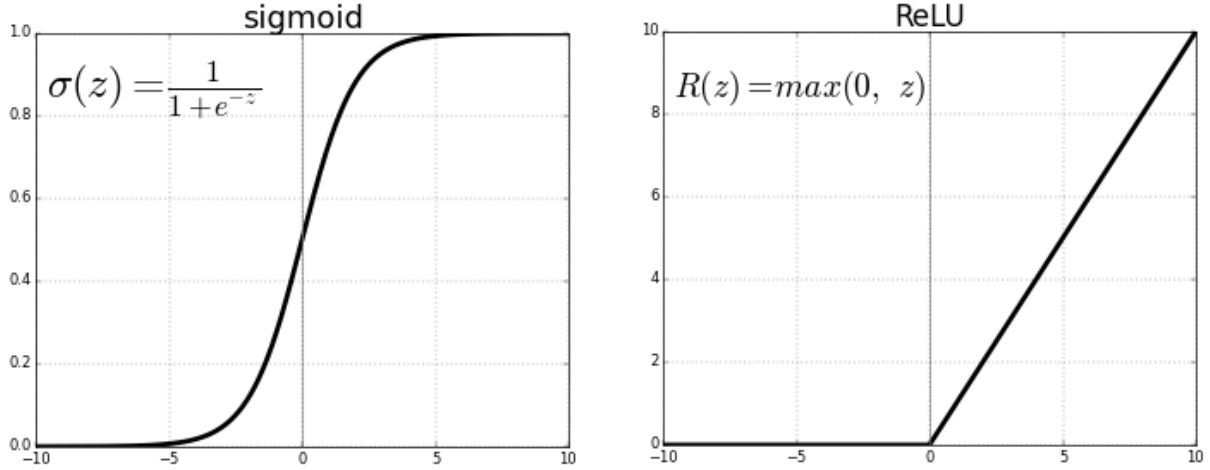
حيث  $x$  تمثل الإشارة الواردة إلى التابع. يتميز تابع سيجمويد بقدرته على تحويل أي قيمة إلى نطاق محدد بين 0 و 1، مما يجعله مناسباً لتفسير الخرج كاحتمال. يُستخدم بشكل واسع في الطبقات الخفية وفي طبقة الخرج لنماذج الشبكات العصبية في مسائل التصنيف الثنائية. ومع ذلك، يُعتبر تابع سيجمويد عرضة لمشكلة تلاشي التدرجات (Vanishing Gradient)؛ حيث تتناقص تدرجاته بشكل كبير مع اقتراب القيم من 0 أو 1، مما يؤدي إلى بطء عملية التعلم، خصوصاً عند استخدام خوارزميات تعتمد على التدرج لضبط الأوزان في الشبكات العصبية العميقة.

### 2.1.3.2 - تابع الوحدة الخطية المصححة (Rectified Linear Unit - ReLU)

لمعالجة مشكلة تلاشي التدرجات التي يواجهها تابع سيجمويد، يُستخدم تابع الوحدة الخطية المصححة (ReLU) كبديل فعال. يتميز تابع ReLU بكونه خطياً للقيم الموجبة وصفرًا للقيم السالبة، ويُعرف كالتالي:

$$ReLU(x) = \max(0, x)$$

تابع ReLU هو من توابع التنشيط الأكثر شيوعاً في الشبكات العصبية، ويستخدم بشكل خاص في الطبقات الخفية. أثبت هذا التابع كفاءته العالية في تمكين الشبكات من تعلم الأنماط المعقدة بدون تكلفة حسابية كبيرة، مع السيطرة على مشكلة تلاشي التدرجات. تشير العديد من الدراسات إلى أن استخدام تابع ReLU في الشبكات العصبية التلافيفية (Convolutional Neural Networks) يؤدي إلى تحسين سرعة التدريب بشكل كبير مقارنة باستخدام توابع تنشيط أخرى، مما يجعله الخيار الأمثل في العديد من التطبيقات العملية في مجال التعلم العميق. يوضح الشكل 5 الفرق بين تابعي التنشيط sigmoid و ReLU.



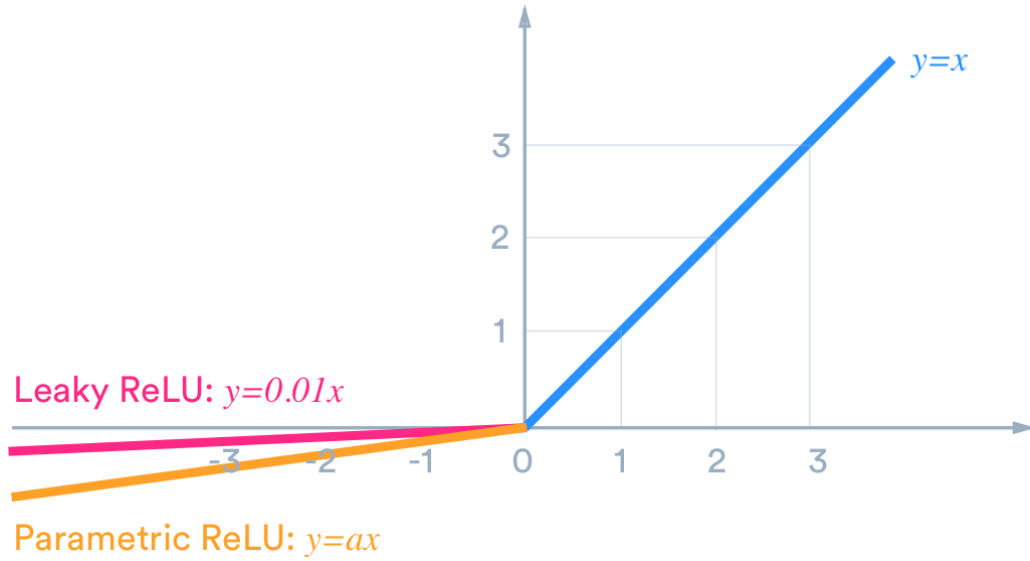
الشكل 5 الفرق بين تابعي التنشيط sigmoid و ReLU

### 3.1.3.2 - تابع الوحدة الخطية المصححة المتسربة (Leaky ReLU)

تم تطوير تابع **Leaky ReLU** كنسخة محسنة من تابع **ReLU** بهدف حل مشكلة تدهور **ReLU (Dying ReLU)** ، التي تحدث عندما تكون مشتقة تابع **ReLU** تساوي صفرًا للقيم السالبة، مما يؤدي إلى عدم تحديث الأوزان أثناء عملية الانتشار الخلفي. في هذه الحالة، قد تصبح بعض الخلايا "ميتة" بالكامل، حيث لا يتم تحديث الأوزان الداخلة إليها أبدًا، مما يجعلها غير نشطة وغير قادرة على التعلم. لحل هذه المشكلة، يقوم **Leaky ReLU** بإضافة مقدار صغير من الميل للقيم السالبة كما هو موضح في الشكل 6. تابع التنشيط **Leaky ReLU**، مما يسمح لبعض التدرجات بالتدفق عبر الخلايا حتى عندما تكون المدخلات سالبة، ويعطى بالعلاقة التالية:

$$Leaky ReLU(x) = \max(x, ax)$$

حيث  $a$  هو ثابت صغير يحدد مقدار الميل المسموح به للقيم السالبة. هذا التعديل يسمح لتابع **Leaky ReLU** بالتعامل بشكل أفضل مع مشكلة الخلايا الميتة، مما يحسن من فعالية التعلم في الشبكات العصبية، خاصة عند التعامل مع نماذج كبيرة ومعقدة.



الشكل 6 تابع التنشيط Leaky ReLU

## 2.3.2- توابع الخسارة Loss Functions

تابع الخسارة أو الكلفة هو أداة تستخدم لتقييم أداء النموذج (بما في ذلك مجموعة الأوزان المحسوبة) أثناء عملية التعلم. يعتمد اختيار تابع الخسارة المناسب على طبيعة المسألة وخرج الطبقة الأخيرة من الشبكة العصبية. سنستعرض فيما يلي أشهر أنواع المسائل وتوابع الخسارة المستخدمة فيها.

### 1.2.3.2- مسائل الانحدار (Regression Problems)

مسائل الانحدار هي تلك التي يكون الهدف منها التنبؤ بقيمة حقيقية أو مستمرة. فيما يلي توابع الخسارة الشائعة المستخدمة في هذا النوع من المسائل:

- تابع متوسط الخسارة المربعة (Mean Squared Error – MSE):

يقيس هذا التابع متوسط مربع الفروقات بين القيمة الحقيقية والقيمة المقدرة باستخدام النموذج:

$$\mathcal{L}(\theta) = \frac{1}{n} \sum_{i=1}^n (y^{(i)} - \hat{y}^{(i)})^2$$

يُعد تابع MSE من الأكثر استخدامًا في مسائل الانحدار بسبب خواصه الرياضية. يوفر هذا التابع مشتقًا متغيرًا يعتمد على قيمة الخسارة؛ حيث تكون المشتقات كبيرة عند الأخطاء الكبيرة، مما يساعد في تقارب النموذج بشكل أسرع للوصول إلى دقة عالية. ومع ذلك، يُعد MSE حساسًا للقيم المتطرفة، حيث يؤدي تربيع الخطأ إلى زيادة الخسارة بشكل أسرع مع زيادة حجم الخطأ، مما يجعل النموذج يولي اهتمامًا أكبر لهذه القيم.

- تابع متوسط الخسارة المطلقة (Mean Absolute Error – MAE):

يقيس هذا التابع متوسط القيمة المطلقة للفروقات بين القيمة الحقيقية والقيمة المقدرة:

$$\mathcal{L}(\theta) = \frac{1}{n} \sum_{i=1}^n |y^{(i)} - \hat{y}^{(i)}|$$

يُستخدم MAE على نطاق واسع في مسائل الانحدار لأنه أكثر مقاومة للقيم المتطرفة مقارنة بـ MSE. لا يزيد MAE بسرعة مع زيادة الخطأ، مما يجعله خيارًا جيدًا عندما تحتوي بيانات التدريب على قيم متطرفة. لكن، نظرًا لأنه ليس قابلاً للاشتقاق عند الصفر، وقد يؤدي ذلك إلى تقليل كفاءة النموذج أثناء التدريب.

### 2.2.3.2 - مسائل التصنيف (Classification Problems)

مسائل التصنيف هي التي تهدف إلى التنبؤ بقيمة صحيحة تمثل فئة معينة. تشمل توابع الخسارة المستخدمة في هذه المسائل:

- تابع الصفر أو الواحد (Zero-One Loss):

يُعطي هذا التابع خسارة قدرها واحد إذا لم تتطابق القيمة المتوقعة مع القيمة الحقيقية، وصفرًا إذا كانت متطابقة:

$$L_{\text{ZeroOne}}(f(\mathbf{x}), y) = \begin{cases} 1 & \text{if } f(\mathbf{x}) \cdot y < 0 \\ 0 & \text{otherwise} \end{cases}$$

لا يأخذ تابع Zero-One Loss درجة الخطأ في الاعتبار وهو غير محدد، لذلك نادرًا ما يُستخدم في تدريب النماذج العصبية.

- تابع الانتروبيا المتقاطعة (Cross Entropy):

يقيس هذا التابع الفارق بين التوزيع الاحتمالي للخروج الفعلي والتوزيع الاحتمالي للخروج المتوقع:

$$\mathcal{L}(\theta) = -\frac{1}{N} \sum_{i=1}^N \sum_{k=1}^K \left[ y_k^{(i)} \log \hat{y}_k^{(i)} + (1 - y_k^{(i)}) \log (1 - \hat{y}_k^{(i)}) \right]$$

يُستخدم تابع الانتروبيا المتقاطعة بشكل واسع في الشبكات العصبية لتفادي التعلم البطيء في طبقات الخرج، حيث يقلل من الفروقات بين التوزيعات الاحتمالية، مما يؤدي إلى تقارب أسرع للنموذج.

هناك نوعان آخران من توابع الانتروبيا المتقاطعة:

○ **تابع الانتروبيا المتقاطعة التصنيفي: (Categorical Cross Entropy)** يُستخدم عند ترميز القيم الفعلية

باستخدام طريقة الترميز "واحد-ساخن" (One-Hot Encoding)

○ **تابع الانتروبيا المتقاطعة المتناثر: (Sparse Categorical Cross Entropy)** يُستخدم عندما يتم ترميز

القيم الفعلية كأعداد صحيحة.

يُعد اختيار تابع الخسارة المناسب أمراً أساسياً لتحسين أداء النموذج وضمان دقة التنبؤات.



### 3- الفصل الثالث

## الدراسة المرجعية

سنبدأ هذا الفصل باستعراض تقنيات التعلم العميق المستخدمة في فهم سلوك المستخدم بشكل عام، ومن ثم سنستعرض الأعمال التي تم طرحها في هذا المجال.

#### 1.3- استخدام التعلم العميق في فهم سلوك المستخدم

أصبح استخدام التعلم العميق ضروريًا لفهم سلوك المستخدم من خلال تفاعله مع المنتجات في أنظمة التوصية الإلكترونية. تستفيد هذه الأنظمة من الشبكات العصبية التكرارية (RNN) لقدرتها الفائقة على معالجة التسلسلات الزمنية المعقدة التي تمثل تفاعلات المستخدم مع المنتجات المختلفة. تعتمد الشبكات العصبية التكرارية على فهم تسلسل سلوك المستخدم من خلال تحليل الأنماط الزمنية والتغيرات في الاهتمامات والاحتياجات قصيرة المدى. على سبيل المثال، يمكن للشبكات العصبية التكرارية مثل LSTM (ذاكرة طويلة قصيرة المدى) تحسين نمذجة تفضيلات المستخدم قصيرة المدى عبر التحكم الديناميكي في انتقال الحالة بين الخطوات الزمنية.

بالإضافة إلى ذلك، يتيح التعلم العميق استخدام تقنيات متقدمة مثل الآليات المبنية على الانتباه (Attention Mechanisms) لتحديد العناصر الأكثر أهمية في سلسلة تفاعل المستخدم، مما يمكن النظام من التركيز على الإجراءات الأكثر تأثيرًا في تحديد التوصيات المستقبلية. من خلال دمج تفضيلات المستخدم طويلة المدى وقصيرة المدى باستخدام إطار عمل تكاملي يعتمد على الانتباه، يمكن تحسين دقة وفعالية التوصيات المقدمة. التجارب المكثفة التي أجريت على مجموعات بيانات عامة وصناعية أكدت فعالية هذه الأساليب، حيث أظهرت النتائج أن النموذج المقترح يفوق العديد من الأساليب الحديثة في القدرة على تقديم توصيات دقيقة ومخصصة. من خلال هذه التطبيقات، يوفر التعلم العميق إطارًا قويًا لفهم السلوك المعقد للمستخدمين وتقديم تجارب تسوق مخصصة تعزز من رضاهم وولائهم.

### 2.3- أنظمة التوصية الأكثر شيوعاً في الوقت الحاضر

- النماذج المستندة إلى المحتوى (Content-Based Models):

تعتمد هذه النماذج على خصائص العناصر التي تم تفاعل المستخدم معها سابقاً. يتم استخدام هذه الخصائص لتقديم توصيات لعناصر جديدة تحمل صفات مشابهة. فعلى سبيل المثال، إذا أبدى المستخدم اهتماماً بكتاب عن التكنولوجيا، فسيتم اقتراح كتب أخرى تتعلق بالتكنولوجيا.

- نماذج التصفية التعاونية (Collaborative Filtering Models):

تعتمد هذه النماذج على جمع تفضيلات العديد من المستخدمين لاكتشاف الأنماط المشتركة في سلوكهم. يتم تقديم التوصيات للمستخدم بناءً على ما أعجب مستخدمين آخرين ذوي اهتمامات مشابهة. تقسم هذه النماذج إلى نوعين: التصفية التعاونية المستندة إلى المستخدم (User-Based) والمستندة إلى العنصر (Item-Based).

- النماذج الهجينة (Hybrid Models):

تجمع هذه النماذج بين مزايا النماذج المستندة إلى المحتوى ونماذج التصفية التعاونية لتعزيز دقة التوصيات. يتم ذلك عن طريق استخدام عدة تقنيات في نفس النموذج، مما يتيح الاستفادة من تفضيلات المستخدمين والخصائص المشتركة للعناصر في آن واحد.

### 3.3- نماذج متقدمة في أنظمة التوصية

في مجال أنظمة التوصية، هناك العديد من النماذج المتقدمة التي تم تطويرها لتحسين فهم سلوك المستخدمين وتقديم توصيات أكثر دقة وملاءمة. تتنوع هذه النماذج في الطرق التي تعتمد عليها لتعلم الأنماط وتقديم التوصيات، وتشمل بعض النماذج الحديثة البارزة ما يلي:

- **ASVD [Koren, 2008]:** يمثل هذا النموذج المستخدمين بناءً على العناصر التي تفاعلوا معها. يتم تمثيل العناصر بشكل متساوٍ، مما يعني أن كل عنصر يساهم بنفس القدر في تحديد تفضيلات المستخدم. تم تحسين هذا النموذج باستخدام آلية انتباه (attentive mechanism) في النسخة المعروفة بـ A2SVD، حيث يتم تخصيص أهمية لكل عنصر بناءً على دوره في تفاعل المستخدم.

- **DIN [Zhou et al., 2018]:** يستخدم هذا النموذج آلية انتباه ديناميكية لتنشيط العناصر المرتبطة في تاريخ تفاعل المستخدم وفقًا للعنصر المستهدف. هذه الآلية تسمح للنموذج بالتركيز على العناصر ذات الصلة عند تقديم التوصيات، مما يعزز من دقة التنبؤات.
- **LSTM:** يعد نموذج LSTM الكلاسيكي من النماذج الشائعة للتنبؤ المتسلسل. يستخدم هذا النموذج الذاكرة الطويلة قصيرة المدى لالتقاط الأنماط الزمنية في بيانات المستخدمين، مما يساعد في فهم تفضيلات المستخدمين قصيرة المدى بشكل أفضل.
- **NARM [Li et al., 2017]:** هو نموذج للتوصية العصبية يعتمد على آلية الانتباه، حيث يقوم بالتقاط الغرض الرئيسي للمستخدم في الجلسة الحالية من خلال دمج آلية الانتباه في إطار الشبكات العصبية التكرارية (RNN). يساعد هذا النموذج في التركيز على الهدف الفوري للمستخدم بناءً على سلوكه الأخير.
- **RRN [Wu et al., 2017]:** يستفيد هذا النموذج من LSTM لالتقاط الديناميكيات إلى جانب التفكيك منخفض الرتبة (low-rank factorization). يتميز هذا النموذج عن LSTM التقليدي بتجميع تفاعلات المستخدم-العنصر بناءً على خطوة زمنية محددة، سواء يوميًا أو أسبوعيًا، مما يساعد في تحسين دقة التوصيات بناءً على تفاعلات المستخدم على مدى فترات زمنية مختلفة.
- **LSTM++:** يجمع هذا النموذج بين A2SVD و LSTM لالتقاط كل من التفضيلات طويلة المدى وقصيرة المدى. يُعرف هذا النموذج باسم LSTM++ لتسهيل الإشارة إلى تكاملهما.
- **CA-RNN [Liu et al., 2016]:** هي شبكة عصبية تكرارية تعتمد على السياق، حيث تستخدم مصفوفات إدخال وتحولات خاصة بالسياق في إطار عمل RNN. يتيح هذا النموذج التكيف بشكل أفضل مع السياقات المختلفة للمستخدمين، مما يزيد من دقة التوصيات المقدمة.
- **T-LSTM [Zhu et al., 2017]:** يضيف هذا النموذج إلى LSTM بوابات زمنية (time gates) لنمذجة الفواصل الزمنية بين التفاعلات. يختلف عن النماذج الأخرى بأنه لا يتعامل مع مدة الفاصل الزمني ولكنه يعتمد على البوابات الزمنية لالتقاط كل من الاهتمامات طويلة وقصيرة المدى.

• **DIEN [Zhou et al., 2019]** يعد هذا النموذج من أحدث الأعمال في مجال التوصيات، حيث يقوم بنمذجة

السلوكيات المتسلسلة للمستخدمين باستخدام طبقتين من وحدات GRU. يتكون من مكونين رئيسيين: طبقة استخراج الاهتمامات وطبقة تطوير الاهتمامات، معتمداً على ASVD كمكون طويل المدى.

تعتمد فعالية هذه النماذج على قدرتها على التعلم من البيانات وتحسين دقة التوصيات بناءً على تحليل دقيق لسلوك المستخدمين. تستخدم هذه النماذج تقنيات متقدمة مثل الشبكات العصبية وآليات الانتباه لتقديم توصيات مخصصة تلبي احتياجات المستخدمين وتفضيلاتهم الفريدة.

Model	Entire	Elec	Movies	CDs	Ads
ASVD	0.8060	0.7727	0.8156	0.8863	0.6463
A <sup>2</sup> SVD	0.8204	0.7838	0.8263	0.9032	0.6501
DIN	0.8293	0.7927	0.8388	0.9111	0.6520
LSTM	0.8272	0.7859	0.8414	0.9086	0.6527
LSTM++	0.8306	0.8028	0.8495	0.9113	0.6535
NARM	0.8290	0.7876	0.8448	0.9130	0.6531
RRN <sub>day</sub>	0.8260	0.7864	0.8406	0.9078	0.6531
RRN <sub>week</sub>	0.8250	0.7869	0.8390	0.9069	0.6500
CARNN	0.8278	0.8106	0.8527	0.9096	0.6551
T-LSTM	0.8387	0.8212	0.8660	0.9181	0.6597
DIEN	0.8361	0.7904	0.8438	0.9128	0.6610

الجدول 1 مقارنة الأداء بحسب AUC

Model	Entire	Elec	Movies	CDs	Ads
ASVD	0.7427	0.7255	0.7539	0.8128	0.3242
A <sup>2</sup> SVD	0.7538	0.7264	0.7565	0.8264	0.3270
DIN	0.7599	0.7349	0.7660	0.8348	0.3320
LSTM	0.7556	0.7311	0.7683	0.8325	0.3296
LSTM++	0.7591	0.7448	0.7742	0.8352	0.3338
NARM	0.7565	0.7323	0.7706	0.8375	0.3327
RRN <sub>day</sub>	0.7550	0.7318	0.7685	0.8322	0.3313
RRN <sub>week</sub>	0.7545	0.7327	0.7676	0.8315	0.3313
CARNN	0.7584	0.7519	0.7763	0.8336	0.3292
T-LSTM	0.7591	0.7448	0.7742	0.8352	0.3338
DIEN	0.7632	0.7327	0.7755	0.8374	0.3343

الجدول 2 مقارنة الأداء بحسب F1-score

## 4-الفصل الرابع

# الأدوات المستخدمة

سوف نشرح في هذا القسم عن الأدوات البرمجية المستخدمة، مع إيضاح المساهمة التي قدمتها كل أداة في تنفيذ القسم العملي.

### • Python

Python هي لغة برمجة عالية المستوى تُستخدم على نطاق واسع في علوم البيانات وتعلم الآلة بفضل بساطتها وسهولة قراءتها. تحتوي Python على مجموعة غنية من المكتبات مفتوحة المصدر مثل TensorFlow وPandas، التي تسهل عمليات تحليل البيانات، وبناء النماذج، وأتمتة العمليات. تعتبر Python الخيار الأمثل للمشاريع المتعلقة بالتعلم الآلي وتطوير الأنظمة المعقدة.

### • PHP

PHP هي لغة برمجة نصية تُستخدم بشكل رئيسي لتطوير تطبيقات الويب. تُعد PHP لغة قوية لتطوير الخوادم وتكامل قواعد البيانات، وتعتبر الخيار الأمثل لتطوير منصات التجارة الإلكترونية وإدارة المحتوى. تُستخدم PHP في بناء الواجهات البرمجية (APIs) والتكامل مع الأنظمة الأخرى، مما يجعلها ضرورية لتطوير الإضافات المخصصة لأنظمة مثل OpenCart.

### • Visual Studio Code (VSCode)

Visual Studio Code هو محرر نصوص مفتوح المصدر من تطوير مايكروسوفت، يُستخدم على نطاق واسع بين المطورين لكتابة وتصحيح الأكواد. يدعم VSCode العديد من لغات البرمجة ويوفر ميزات مثل تكامل Git، وإدارة الإضافات، والتطوير المستند إلى الحاويات (Containers)، مما يجعله أداة فعالة للتطوير البرمجي الشامل.

### • OpenCart

OpenCart هو نظام إدارة محتوى مفتوح المصدر مصمم خصيصًا لبناء متاجر التجارة الإلكترونية. يوفر OpenCart بنية تحتية مرنة وقابلة للتخصيص، مما يسهل دمج الإضافات والتكامل مع خدمات الطرف الثالث. يعد OpenCart خيارًا ممتازًا للمطورين الذين يرغبون في إنشاء وتخصيص منصات تجارة إلكترونية لتلبية احتياجات عملائهم.

## • **Recommenders**

مكتبة Recommenders هي مكتبة مفتوحة المصدر من مايكروسوفت مصممة خصيصًا لبناء وتقييم أنظمة التوصية. توفر المكتبة مجموعة من الأدوات والخوارزميات المتقدمة لتطوير أنظمة توصية مخصصة، مثل التوصية التعاونية والترشيح التعاوني القائم على المحتوى.

## • **TensorFlow**

TensorFlow هو مكتبة مفتوحة المصدر من تطوير جوجل، تستخدم بشكل واسع في بناء نماذج تعلم الآلة والتعلم العميق. توفر TensorFlow بنية تحتية مرنة وفعالة للتدريب على البيانات الكبيرة وتطبيق الشبكات العصبية المعقدة مثل الشبكات العصبية التكرارية (RNN) والشبكات العصبية التلافيفية (CNN).

## • **Pandas**

Pandas هي مكتبة برمجية مفتوحة المصدر بلغة Python، توفر هياكل بيانات وأدوات تحليلية عالية الأداء. يتيح Pandas معالجة البيانات بسهولة من خلال عمليات التحليل والتنظيف والتحضير، مما يجعلها أداة أساسية في مشاريع علوم البيانات وتعلم الآلة. توفر المكتبة إمكانيات متقدمة للعمل مع البيانات المهيكلة مثل الجداول الزمنية والإحصائيات.

## • **MLflow**

MLflow هو نظام مفتوح المصدر لإدارة دورة حياة تعلم الآلة. يتيح MLflow تتبع وتنظيم تجارب النماذج، إدارة بيانات النماذج، ونشر النماذج لخدمة الإنتاج. تم استخدامها في عملية توظيف النموذج.

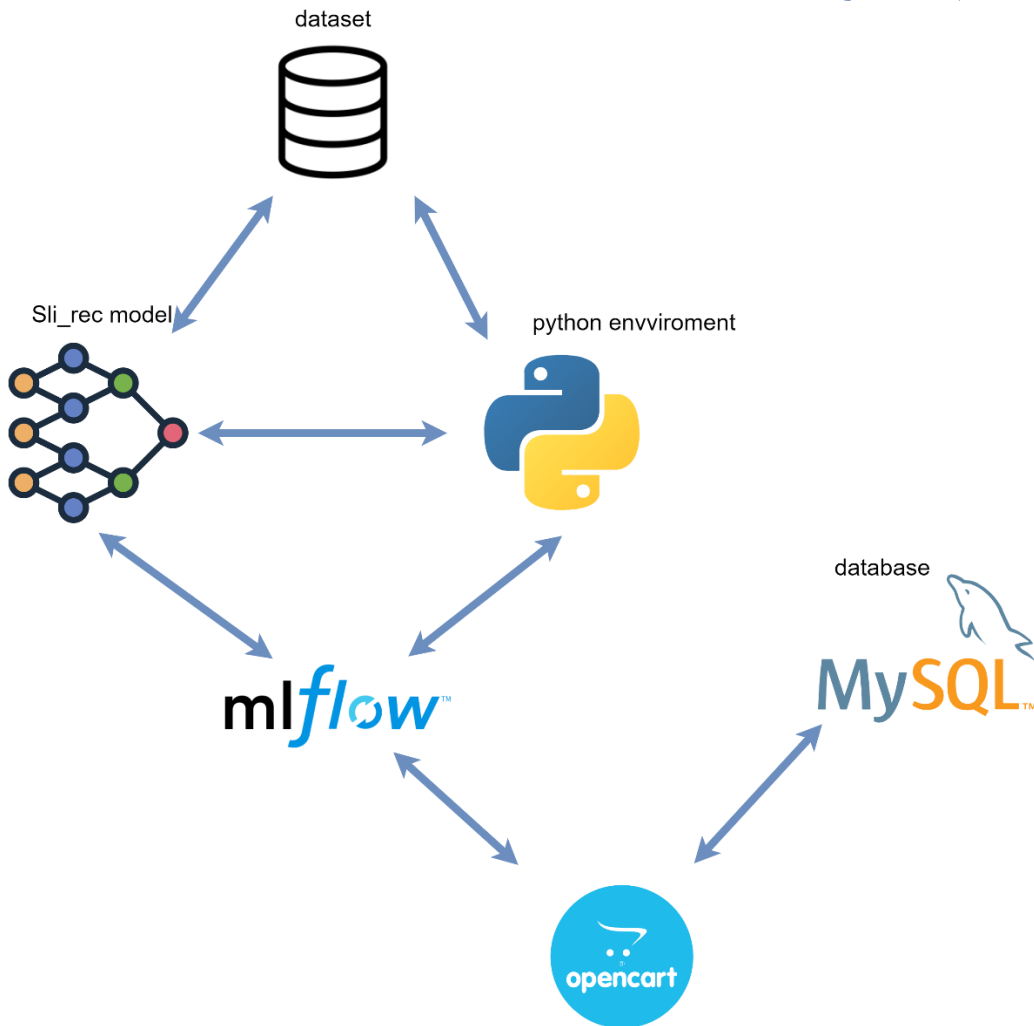
تعتبر هذه الأدوات مجتمعةً أساسية في تطوير وتنفيذ المشروع، حيث توفر بيئة متكاملة لإنشاء نماذج تعلم الآلة وتحليل البيانات وتطبيقها في نظام إدارة المحتوى.

## 5- الفصل الخامس

# القسم العملي

في هذا الفصل، سيتم تناول الخطوات العملية التي تم اتباعها لتطوير نظام توصيات للتجارة الإلكترونية باستخدام نموذج ذكاء اصطناعي. يشمل القسم العملي عدة مراحل تبدأ من إعداد بيئة العمل، وإنشاء النموذج، وتدريبه على بيانات محددة، ومن ثم ربطه مع منصة التجارة الإلكترونية *OpenCart*.

### 1.5- الهيكل العام للمشروع







الشكل 7 الهيكل العام للمشروع

## 2.5- إعداد بيئة العمل

في البداية، تم تحميل الرمز البرمجي لمنصة OpenCart وتثبيتها لتجهيز بيئة العمل المناسبة.

مراحل الإعداد تضمنت:

- تحميل حزمة OpenCart من الموقع الرسمي.
  - إعداد قاعدة بيانات MySQL وربطها بـ OpenCart.
- حيث تأتي البنية التحتية لمنصة OpenCart مع قاعدة بيانات جاهزة، وقد تم إضافة جدول في قاعدة البيانات نحفظ به بتفاعلات المستخدمين مع المنتجات.
- بنية الجدول المضاف كما في الشكل 8 جدول تفاعل المستخدمين مع المنتجات

#	Name	Type
1	interaction_id 	int(11)
2	user_id 	int(11)
3	product_id 	int(11)
4	category_id 	int(11)
5	interaction_timestamp	datetime
6	interaction_type	varchar(50)

الشكل 8 جدول تفاعل المستخدمين مع المنتجات

- تشغيل الموقع محليًا للتأكد من ضمان سير العمل.

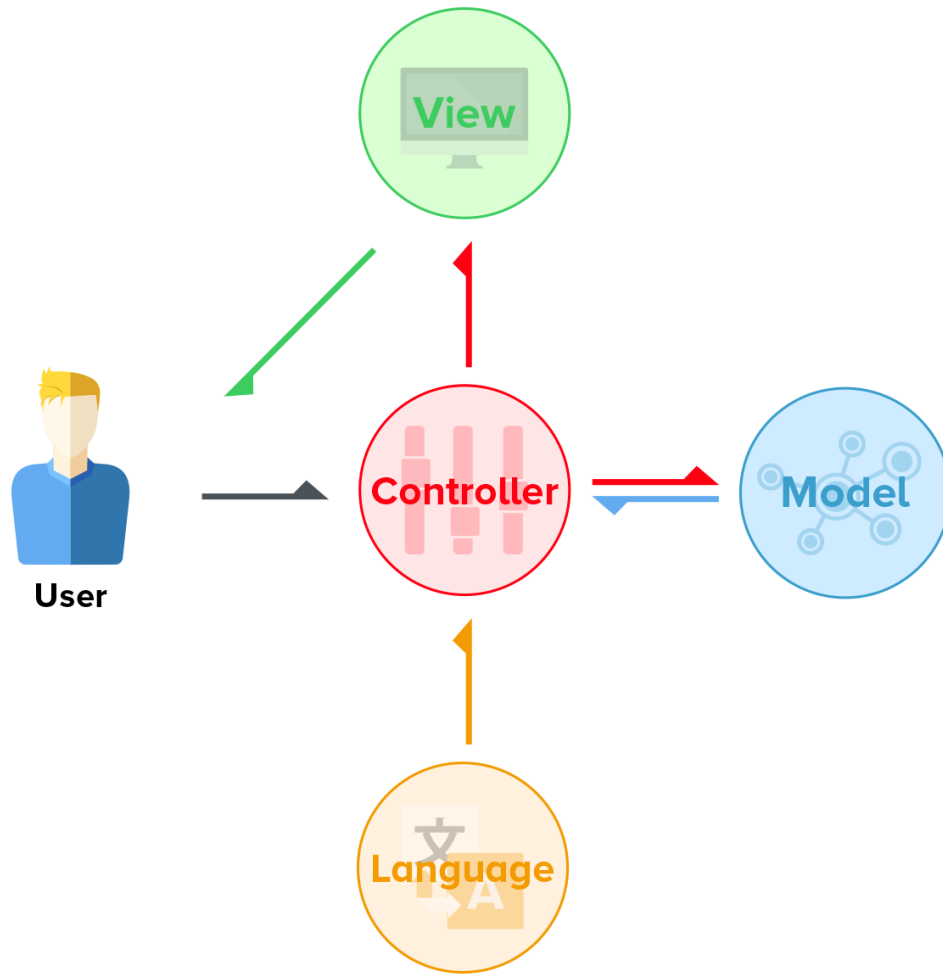
## 3.5- إنشاء الإضافة (Extension) الأساسية

بعد تثبيت OpenCart وتشغيل الموقع بنجاح، تم إنشاء إضافة (Extension) أساسية لتكون ركيزة التعامل مع نموذج التوصية. تم ربط هذه الإضافة بعدة أحداث (events) تقوم بتسجيل تفاعلات المستخدم مع المنتجات وحفظ هذه التفاعلات في جدول الذي تم إضافته على قاعدة البيانات.

إن OpenCart يعتمد MVCL design pattern الموضح في الشكل 9 MVCL design pattern.

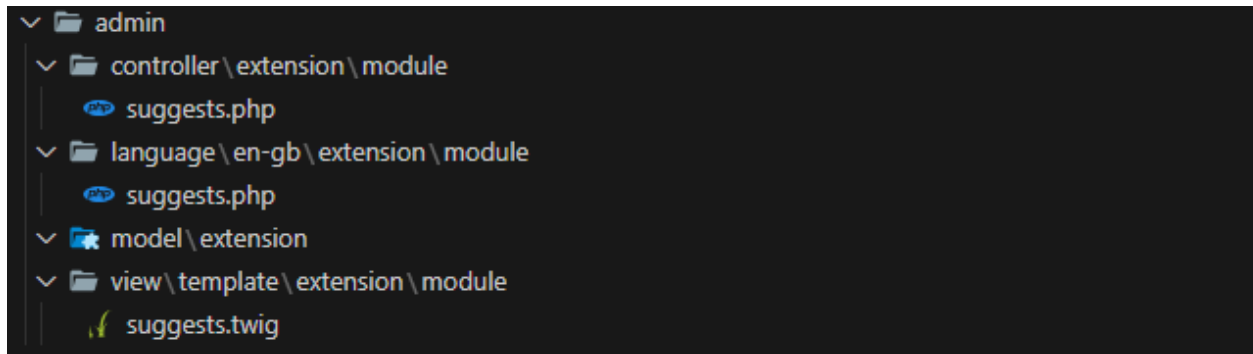
إن بنية الملفات في OpenCart مقسمة إلى قسمين، قسم مخصص لمدير الموقع وقسم مخصص لواجهة المستخدم.





الشكل 9 MVCL design pattern

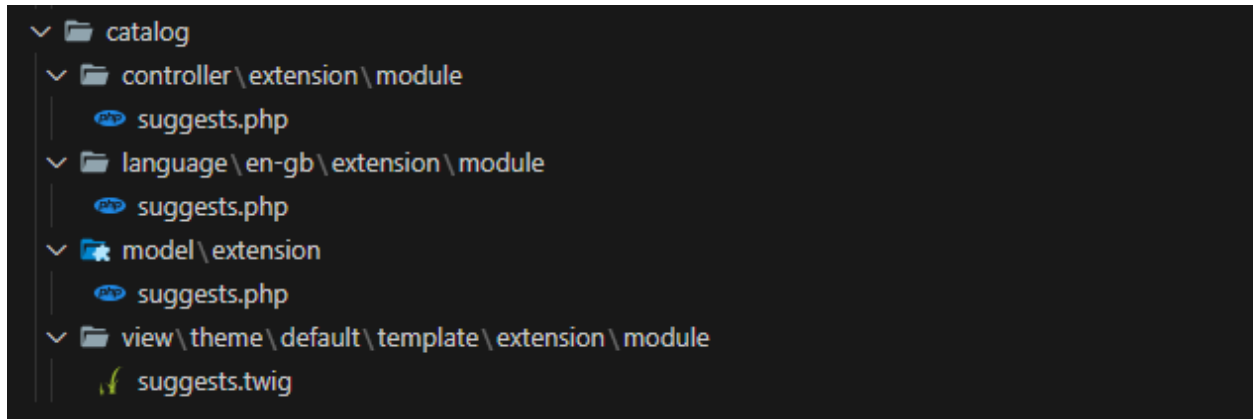
في القسم المخصص للمدير تم إنشاء ملفات وفق الهرمية الموضحة في الشكل 10 هرمية الملفات في قسم المدير.



الشكل 10 هرمية الملفات في قسم المدير

حيث تم تصميم الواجهة التي تظهر لمدير الموقع والاعدادات التي يتيحها له، تم إضافة خيارات بسيطة مثل عدد النتائج التي سيقترحها النموذج والحجم الذي ستظهر به المنتجات المقترحة للزبون.

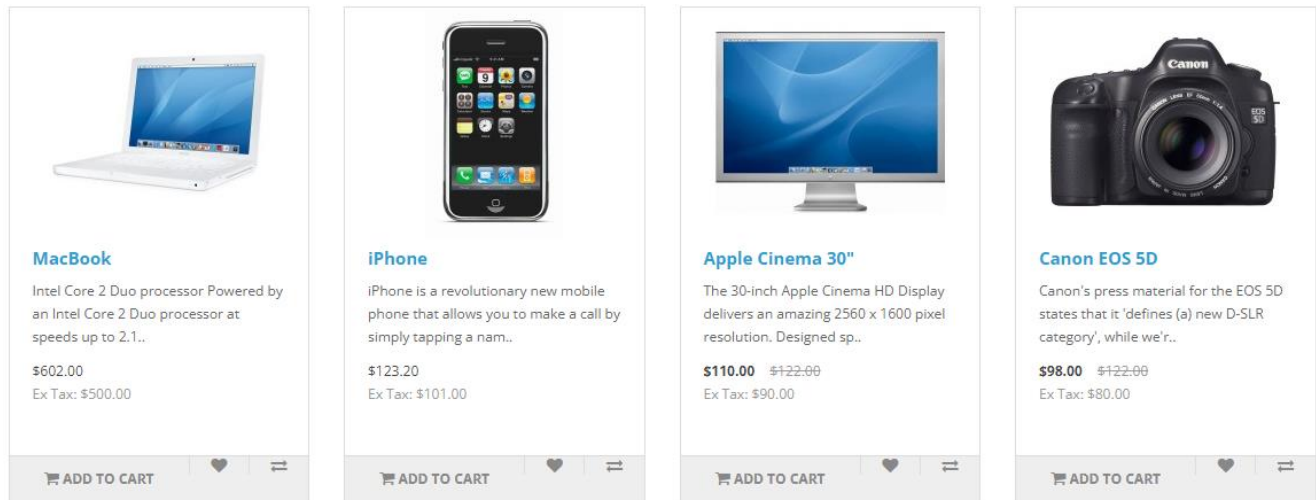
أما القسم المخصص لوجهات المستخدم فيكون وفق الهرمية الموضحة في الشكل 11 هرمية الملفات في قسم المستخدم.



الشكل 11 هرمية الملفات في قسم المستخدم

تم تصميم واجهة المستخدم وإنشاء model يقوم بإعطاء نتائج عشوائية بدايةً ليتم ربطه لاحقاً بالنموذج المختار. وتظهر المنتجات المقترحة للمستخدم في الصفحة الرئيسية كما في الشكل 12 كيفية ظهور الاقتراحات للمستخدم:

#### Suggests



الشكل 12 كيفية ظهور الاقتراحات للمستخدم

## 4.5- البحث عن نماذج التوصية

تم إجراء بحث مكثف حول النماذج المختلفة المتاحة للتوصية في أنظمة التجارة الإلكترونية، وأوجدنا المكاتب التالية:

- **Microsoft Recommenders:**

تعتبر مكتبة Microsoft Recommenders مكتبة مفتوحة المصدر تم تطويرها بواسطة فريق Microsoft Azure. توفر المكتبة مجموعة واسعة من خوارزميات التوصية، بما في ذلك التصفية التعاونية (Collaborative Filtering) والشبكات العصبية المتكررة (RNN) والشبكات العصبية التلافيفية (Convolutional Neural Networks - CNN)، بالإضافة إلى نماذج هجينة تجمع بين أساليب متعددة.

- **Surprise:**

مكتبة Surprise هي مكتبة مفتوحة المصدر مكتوبة بلغة Python، وتستخدم لبناء نماذج التصفية التعاونية وتحليلها. تركز المكتبة على النماذج القائمة على المصفوفة، مثل التفكيك المفرد للقيم (Singular value decomposition) وخوارزميات (K-Nearest Neighbors (KNN).

- **LightFM:**

مكتبة LightFM هي مكتبة مخصصة لبناء نماذج التوصية الهجينة التي تجمع بين التصفية التعاونية والنماذج المستندة إلى المحتوى (Content-Based Models). تعتمد المكتبة على خوارزمية التعلم العميق لتعلم التفاعلات المعقدة بين المستخدمين والعناصر.

- **TensorFlow Recommenders (TFRS):**

مكتبة TensorFlow Recommenders (TFRS) هي مكتبة مفتوحة المصدر مبنية على إطار عمل TensorFlow من Google، ومخصصة لبناء نماذج توصية باستخدام تقنيات التعلم العميق. تتيح المكتبة للمستخدمين بناء نماذج توصية متطورة باستخدام مجموعة متنوعة من الأساليب مثل التعلم المعزز (Reinforcement Learning) والشبكات العصبية العميقة (Deep Neural Networks).

- **TorchRec:**

PyTorch هي مكتبة مفتوحة المصدر من فريق PyTorch مخصصة لبناء نماذج توصية باستخدام التعلم العميق. تقدم TorchRec أدوات لبناء نماذج توصية مخصصة ومعقدة باستخدام بنية PyTorch المرنة، مع دعم واسع من المجتمع وتكامل مع بيئات التطوير المختلفة.

المكتبة	المزايا	العيوب
<b>Microsoft Recommenders</b>	<ul style="list-style-type: none"> <li>- تحتوي على مجموعة متنوعة من النماذج (تصفية تعاونية، تعلم عميق، نماذج هجينة).</li> <li>- الوصول إلى خوارزميات متقدمة.</li> <li>- إرشادات حول تطوير نظام توصية فعال.</li> <li>- توفر أدوات لتحليل أداء النماذج وتحسينها.</li> </ul>	<ul style="list-style-type: none"> <li>- تتطلب معرفة قوية بخدمات Azure لاستخدامها بفعالية.</li> <li>- قد تكون معقدة للمبتدئين بسبب تنوع النماذج والخيارات المتاحة.</li> </ul>
<b>Surprise</b>	<ul style="list-style-type: none"> <li>- سهولة الاستخدام والتهيئة للمستخدمين الجدد في مجال التوصية.</li> <li>- تدعم مجموعة متنوعة من خوارزميات التصفية التعاونية.</li> <li>- توفر أدوات تقييم قوية لاختبار أداء النماذج.</li> </ul>	<ul style="list-style-type: none"> <li>- تركز فقط على التصفية التعاونية، ولا تدعم النماذج المستندة إلى المحتوى أو التعلم العميق.</li> <li>- أقل مرونة في دمج تقنيات حديثة مثل آليات الانتباه (Attention Mechanism).</li> </ul>
<b>LightFM</b>	<ul style="list-style-type: none"> <li>- تدعم النماذج الهجينة التي تجمع بين التصفية التعاونية والنماذج المستندة إلى المحتوى.</li> <li>- قادرة على التعامل مع البيانات الصريحة والضمنية.</li> <li>- مناسب للسيناريوهات ذات التفاعلات المحدودة بين المستخدمين والعناصر.</li> </ul>	<ul style="list-style-type: none"> <li>- قد تكون أداؤها أقل في التوصيات المعقدة مقارنة بالنماذج المعتمدة على التعلم العميق.</li> <li>- تتطلب إعدادًا دقيقًا للمعاملات لتحقيق أفضل أداء.</li> </ul>
<b>TensorFlow Recommenders (TFRS)</b>	<ul style="list-style-type: none"> <li>- توفر أدوات قوية لبناء نماذج توصية معقدة باستخدام التعلم العميق.</li> <li>- تكامل سهل مع بيئة TensorFlow وأدواتها الأخرى.</li> <li>- مناسبة للتطبيقات الصناعية والبحثية الكبيرة والمعقدة.</li> </ul>	<ul style="list-style-type: none"> <li>- تحتاج إلى معرفة متقدمة في التعلم العميق وTensorFlow.</li> <li>- تعقيد في الإعداد والتدريب قد يكون تحديًا للمبتدئين.</li> </ul>
<b>TorchRec</b>	<ul style="list-style-type: none"> <li>- مرونة كبيرة في تصميم النماذج باستخدام بنية PyTorch.</li> <li>- دعم واسع من المجتمع وتكامل مع بيئات التطوير المختلفة.</li> <li>- مناسب لنماذج التوصية المتقدمة.</li> </ul>	<ul style="list-style-type: none"> <li>- قد يكون معقدًا في الاستخدام للمبتدئين.</li> <li>- يتطلب معرفة عميقة بتصميم النماذج والتعلم العميق.</li> </ul>

الجدول 3 مقارنة بين المكتبات المتاحة

## 5.5- اختيار Microsoft Recommenders

تم اختيار مكتبة Microsoft Recommenders لتنفيذ نظام التوصية في المشروع لعدة أسباب، وهي كما يلي:

- تنوع النماذج وخوارزميات التوصية:  
هذا التنوع يتيح لنا اختيار النموذج الأمثل الذي يتناسب مع متطلبات المشروع وخصائص البيانات المتاحة.
  - أدوات تحليل وتحسين الأداء:  
تتضمن المكتبة آليات لتجربة النماذج ومقارنة أدائها باستخدام مقاييس تقييم مختلفة مثل الدقة والتذكر (Precision & Recall) و AUC. هذه الأدوات تساعد في تحسين دقة وكفاءة نظام التوصية بمرور الوقت.
  - دعم تقنيات التعلم العميق المتقدمة:  
تحتوي المكتبة على نماذج متقدمة تعتمد على تقنيات التعلم العميق مثل SLi-Rec، الذي يستخدم آليات الشبكات العصبية طويلة المدى (LSTM) وآليات الانتباه (Attention Mechanism) لتقديم توصيات مخصصة ومتكيفة مع تفضيلات المستخدمين الفردية في الوقت الفعلي.
  - إمكانية الوصول إلى خوارزميات متقدمة وأحدث التقنيات:  
هذا يضمن أن النظام المستخدم يستفيد من أحدث الابتكارات في المجال، مما يعزز من فاعليته وكفاءته.
- تحتوي المكتبة على عدد كبير من النماذج يصل إلى 30 نموذج، نذكر أهم النماذج الموجودة التي تم العمل بها:
- Attentive Asynchronous Singular Value Decomposition (A2SVD)
  - Short-term and Long-term Preference Integrated Recommender (SLi-Rec)
  - Multi-Interest-Aware Sequential User Modeling (SUM)
  - Convolutional Sequence Embedding Recommendation (Caser)
  - GRU
- جميع هذه النماذج تعتمد على الخوارزميات المتسلسلة (Sequential-Based Algorithms) التي تهدف إلى التقاط كل من تفضيلات المستخدم الطويلة والقصيرة المدى باستخدام آلية الانتباه (Attention Mechanism). هذه النماذج تعمل على تحسين دقة التوصيات من خلال التعلم من أنماط التفاعل المتتابع للمستخدمين.

## كيف تعمل هذه النماذج؟

- التقاط التفضيلات طويلة المدى:

الخوارزميات المتسلسلة تستخدم بيانات التفاعل السابقة للمستخدم لتكوين فهم عميق للتفضيلات طويلة المدى. على سبيل المثال، إذا كان المستخدم يميل إلى نوع معين من المنتجات على مدى فترة طويلة، فسيتم التقاط هذه الأنماط من خلال النماذج المتسلسلة وتضمينها في عملية التوصية.

- التقاط التفضيلات قصيرة المدى:

بالإضافة إلى ذلك، تقوم هذه النماذج بتحليل سلوك المستخدمين في الجلسات القصيرة أو التفاعلات الأخيرة. هذا يساعد على التقاط التفضيلات المؤقتة التي قد تكون مختلفة عن التفضيلات طويلة المدى، مثل الاهتمام بمنهج معين في موسم خاص أو حدث معين.

- استخدام آلية الانتباه (Attention Mechanism):

آلية الانتباه تعد جزءاً مهماً في هذه النماذج لأنها تسمح للنموذج بالتركيز على أجزاء معينة من بيانات التفاعل بناءً على أهميتها للسياق الحالي. هذا يعني أن النموذج يمكنه تخصيص وزن أعلى للتفاعلات الأخيرة أو لتلك التي تتناسب مع السياق الحالي للمستخدم، مما يحسن دقة التوصيات.

- دمج التفضيلات المتعددة في التوصية:

من خلال الجمع بين التفضيلات طويلة وقصيرة المدى باستخدام آلية الانتباه، يمكن لهذه النماذج تقديم توصيات دقيقة وشخصية للمستخدمين. هذا الدمج يساعد على تقديم توصيات تلبي الاحتياجات الفورية للمستخدمين وتتماشى مع اهتماماتهم العامة.

أحد أهم هذه النماذج هو نموذج **SLi-Rec** ما يميزه هو أنه (time-aware) أما بقية النماذج فتركز فقط على ترتيب التفاعلات التي قام بها المستخدم.

## 6.5- إعداد وتجهيز مجموعة البيانات (Dataset)

### 1.6.5- تنسيق البيانات المدخلة (Input Data Format)

يحتوي ملف البيانات المدخلة على 8 أعمدة مرتبة بالشكل التالي:

- `<label>` هو قيمة ثنائية (binary) حيث يتم تعيين 1 للحالات الإيجابية و 0 للحالات السلبية.
- الحالة الإيجابية تعني تفاعل المستخدم مع العنصر خلال الطابع الزمني، والحالة السلبية تعني عدم تفاعله.
- `<user_id>` يمثل معرف المستخدم.
- `<item_id>` يمثل معرف العنصر أو المنتج.
- `<category_id>` يمثل معرف فئة العنصر.
- `<timestamp>` يعبر عن الطابع الزمني.
- ثم لدينا ثلاثة أعمدة هي `<history_timestamp>` `<history_category_ids>` `<history_item_ids>` تسجل قائمة سلوكيات المستخدم حتى وقت `<timestamp>`.

يتم فصل الأعمدة باستخدام علامة الجدولة `"\t"`.

أما الأعمدة التي تبدأ بـ `<*_history>` يتم فصل العناصر بفواصل.

على سبيل المثال، يمكن أن يكون شكل إحدى الحالات كما في الشكل 13 مثال عن البيانات المدخلة:

```
1 A1QQ86H5M2LVW2 B0059XTU1S Movies 1377561600 B002ZG97WE,B004IK30PA,B000BNX3AU,B0017ANB08,B005LAIHW2
Movies,Movies,Movies,Movies,Movies 1304294400,1304812800,1315785600,1316304000,1356998400
```

الشكل 13 مثال عن البيانات المدخلة

### 2.6.5- معالجة البيانات (data preprocessing)

خلال مرحلة معالجة البيانات المبدئية، تم استخدام سكرت مخصص لمعالجة بيانات توفره المكتبة واسمه `amazon_reviews`، حيث تم التعديل عليه ليناسب البيانات التي لدينا.

البيئة عمل السكريبت كالتالي:

- تحميل البيانات الخام:

في البداية، يتم تحميل البيانات الخام من ملفات نصية تحتوي على مراجعات المستخدمين للمنتجات. هذه البيانات تتضمن معلومات مثل معرف المستخدم (user\_id)، معرف المنتج (item\_id)، تصنيف الفئة (category\_id)، والطابع الزمني (timestamp) لكل تفاعل.

- تنظيف البيانات:

بعد تحميل البيانات، يتم تنظيفها للتأكد من أنها خالية من القيم المفقودة أو التكرارات غير الضرورية. يشمل التنظيف أيضاً إزالة الحالات التي لا تحتوي على الحد الأدنى من المعلومات المطلوبة لتحليل البيانات، مثل نقص معرف المستخدم أو المنتج.

- توليد قواميس ترميز المعرفات (ID Mapping Dictionaries):

يقوم الملف بإنشاء قواميس ترميز لمعالجة وتحويل المعرفات النصية (user\_id، item\_id، category\_id) إلى أرقام صحيحة (integers) تسهل من عملية المعالجة الحسابية للنماذج. يتم إنشاء هذه القواميس باستخدام تابع create\_vocab(train\_file, user\_vocab, item\_vocab, cate\_vocab)\_

هذه القواميس تُستخدم لاحقاً لتحويل بيانات التدريب والتقييم إلى تمثيلات رقمية. يمكن أن تُنشأ هذه القواميس فقط من train\_file، مما يعني أن أي معرفات جديدة تظهر في ملفات التقييم أو الاختبار (valid\_file أو test\_file) سيتم اعتبارها كمعرفات غير معروفة وتُعيّن إلى الفهرس الافتراضي (0).

- تقسيم البيانات إلى حالات إيجابية وسلبية:

واحدة من الخطوات الأساسية في إعداد البيانات هي توليد الحالات الإيجابية والسلبية للتعلم. في أنظمة التوصية وكما ذكرنا الحالة الإيجابية تمثل تفاعل المستخدم مع المنتج، بينما الحالة السلبية تمثل عدم وجود تفاعل (يُفترض ذلك). الملف يسمح بإنشاء عينات سلبية بشكل ديناميكي من خلال تحديد عدد العينات السلبية (num\_negs) التي يجب أن ترافق كل حالة إيجابية في بيانات التدريب.

- إعداد بيانات التسلسل الزمني (Time-based Data Preparation):

نظراً لأن النموذج المختار SLi\_Rec يعتمد على الوقت (Time-Aware Model)، فإن ملف معالجة البيانات يهتم بشكل خاص بإعداد البيانات بحيث يتضمن كل تفاعل الطابع الزمني (timestamp) للتفاعل وأيضاً تاريخ السلوكيات السابقة للمستخدم حتى ذلك الطابع الزمني. يتم تخزين هذه السلوكيات في الأعمدة <history\_item\_ids>, <history\_category\_ids>, <history\_timestamp>.



- تنسيق البيانات لعمليات التدريب والتقييم:  
بعد تجهيز البيانات، يتم ترتيبها بشكل معين بحيث تتوافق مع متطلبات النماذج المختلفة في المكتبة. كل حالة إيجابية تُتبع بعدد محدد من الحالات السلبية (num\_negs). هذا الترتيب مهم جدًا لعمليات التدريب باستخدام خوارزميات التصنيف مثل Softmax، التي تحتاج إلى ترتيب معين للبيانات لحساب احتمالات التوصيات بشكل صحيح.
- حفظ البيانات الجاهزة للاستخدام:  
أخيرًا، بعد الانتهاء من كل عمليات التحضير، يتم حفظ البيانات الجاهزة في تنسيق قابل للاستخدام مباشرة في عمليات التدريب والتقييم للنماذج. يقوم الملف بتوفير مواقع حفظ واضحة للملفات التي تتضمن بيانات التدريب، التقييم، والاختبار، بالإضافة إلى قواميس الترميز (vocabularies).

### 3.6.5- إنشاء بيانات وهمية للتدريب عليها

تتضمن هذه العملية إنشاء منتجات متعددة الفئات، ثم استخدام سكرت يولد بيانات تفاعل المستخدمين مع هذه المنتجات.

- إنشاء منتجات متعددة الفئات  
تم باستخدام أدوات الذكاء الصناعي إنشاء مجموعة من المنتجات تنتمي إلى فئات مختلفة لتعكس تنوع المنتجات في بيئة التجارة الإلكترونية الواقعية. تم تعريف كل منتج بمواصفات معينة، مثل معرف المنتج (item\_id)، واسم المنتج، والفئة (category\_id). تمثل الفئات تصنيفات واسعة للمنتجات، مثل الإلكترونيات، والكتب، والملابس، وما إلى ذلك. تم إعداد مجموعة بيانات تحتوي على عدد كافٍ من المنتجات في كل فئة لتضمن تمثيلًا جيدًا للتنوع ولضمان فعالية النموذج في التعامل مع جميع أنواع المنتجات.
- توليد بيانات التفاعل بين المستخدمين والمنتجات  
بعد إنشاء المنتجات، تم كتابة سكرت لتوليد بيانات تفاعل المستخدمين مع هذه المنتجات بشكل شبه عشوائي. يقوم هذا السكرت بتوليد تفاعلات وهمية بين المستخدمين والمنتجات باستخدام عدة معايير تشمل:  
معرف المستخدم (user\_id)، معرف المنتج (item\_id)، معرف الفئة (category\_id)، الطابع الزمني (timestamp).

السكرت يتبع الخطوات التالية لتوليد بيانات التفاعل:

- تهيئة المستخدمين وقائمة التفاعلات:  
تم إنشاء قائمة من المستخدمين الوهميين (users) عددهم 400 مستخدم، كل مستخدم يحمل معرف فريد مثل "user\_1"، "user\_2"، وهكذا.

- تحديد تسلسلات التفاعل وأوزان الاحتمالات:

تم تحديد مجموعة من تسلسلات التفاعل التي تعكس سلوك المستخدمين الشائع على موقع التجارة الإلكترونية. على سبيل المثال، المستخدم قد "يشاهد" منتجًا فقط أو "يشاهد" المنتج ثم "يضيفه إلى السلة" أو حتى "يراجع" المنتج. تم تعيين أوزان احتمالات (weights) لكل تسلسل لتحديد مدى احتمالية حدوث هذا التسلسل. على سبيل المثال، احتمال أن يقوم المستخدم بـ"مشاهدة" منتج فقط هو 35% بينما احتمال أن يشاهد ويضيف المنتج للسلة ثم يراجع هو 5%، حيث تم وضع الاحتمالات بشكل تقديري.

- إعداد فئات المنتجات:

تم إنشاء قائمة من قوائم المنتجات، حيث تحتوي كل قائمة على مجموعة من المنتجات التي تنتمي إلى فئة معينة (مثل "الإلكترونيات"، "الأجهزة المنزلية"، "الأزياء"، إلخ)، هذه المنتجات هي التي تم إنشاؤها باستخدام أحد أدوات الذكاء الصناعي.

- إنشاء التفاعلات لكل مستخدم:

لكل مستخدم في قائمة المستخدمين:

يتم تعيين base\_date كنقطة بداية جديدة لكل مستخدم. ثم يتم خلط ترتيب الفئات بشكل عشوائي لتوزيع الفئات لكل مستخدم.

لكل قائمة من المنتجات في فئة مختارة:

يتم تحديث التاريخ الأساسي بمدة زمنية عشوائية لإضافة تنوع زمني في التفاعلات. ثم يتم تحديد عدد عشوائي من المنتجات التي سيتفاعل معها المستخدم، ومن ثم يتم اختيار هذه المنتجات بشكل عشوائي من قائمة المنتجات التي تتبع لفئة محددة.

لكل منتج تم اختياره للتفاعل:

يتم اختيار تسلسل تفاعل عشوائي بناءً على الأوزان المحددة مسبقًا. ثم يتم تحديث التاريخ الحالي بإضافة فترات زمنية عشوائية.

مع ملاحظة أن الفترات الزمنية المضافة بين تفاعل المستخدم مع منتجات من نفس الفئة أقصر من الفترة الزمنية المضافة بين تفاعل المستخدم مع فئات مختلفة.

كما أن الفترات الزمنية المضافة بين الإجراءات لنفس المنتج تكون قصيرة جدًا، أي من 10 إلى 20 دقيقة.

لكل إجراء في تسلسل التفاعل:

إذا كان الإجراء "مراجعة"، يتم تعيين تقييم عشوائي بين 1 و 5.

يتم تسجيل التفاعل مع تفاصيل مثل معرف المستخدم والمنتج والفئة ونوع التفاعل والطابع الزمني في قائمة التفاعلات interactions.

## 7.5- تدريب النموذج

تم تدريب نموذج التوصية باستخدام البيانات المعدة مسبقًا. تتضمن التدريب:

- اختيار المعاملات المناسبة للنموذج:

حيث تم اختيار المعاملات التي وضعها فريق عمل Microsoft وهي كالتالي:

- $learning\_rate=0.001$ : معدل التعلم. يحدد حجم الخطوة التي يتخذها النموذج أثناء التحديث التدريجي للأوزان لتقليل الخسارة. قيمة صغيرة (مثل 0.001) تساعد في التعلم البطيء والثابت، مما يقلل من خطر تجاوز النقطة المثلى.
- $dropout=0.3$ : معامل التجاهل. يمثل النسبة المئوية من العقد (neurons) في طبقات الشبكة العصبية التي يتم تجاهلها بشكل عشوائي أثناء كل تكرار (iteration) لتجنب الإفراط في التكيف (overfitting). قيمة 0.3 تعني تجاهل 30% من العقد.
- $item\_embedding\_dim=32$ : بُعد تضمين (embedding) العناصر. يحدد حجم الفضاء الشعاع الذي يتم فيه تمثيل العناصر (items). كل عنصر يتم تمثيله بواسطة شعاع من 32 بُعدًا.
- $cate\_embedding\_dim=8$ : بُعد تضمين الفئات. يشير إلى عدد الأبعاد المستخدمة لتمثيل فئات المنتجات. هنا، كل فئة تمثل بشعاع من 8 أبعاد.
- $l2\_norm=0$ : تنظيم (L2 Regularization). يستخدم لتجنب الإفراط في التكيف عن طريق إضافة عقوبة إلى الخسارة التي تعتمد على حجم أوزان النموذج. هنا القيمة 0 أي أنه لا يتم استخدام تنظيم L2.
- $batch\_size=400$ : حجم الدفعة (batch size). يشير إلى عدد العينات التي يتم تمريرها إلى النموذج في كل تحديث للأوزان. حجم دفعة 400 أي أنه تم استخدام 400 عينة في كل تكرار.
- $epochs = 10$ : عدد التكرارات (epochs). يشير إلى عدد المرات التي يتم فيها تمرير كامل مجموعة البيانات عبر النموذج أثناء التدريب. عدد التكرارات 10 أي أن مجموعة البيانات ستتم معالجتها كاملة 10 مرات.
- $train\_num\_ngs=4$ : عدد العينات السلبية (negative samples) لكل عينة إيجابية أثناء التدريب. أي أن كل عينة إيجابية سيتم إقرانها بـ 4 عينات سلبية لتدريب النموذج.
- $valid\_num\_ngs=4$ : عدد العينات السلبية لكل عينة إيجابية أثناء التحقق (validation). هذا يساعد في تقييم النموذج أثناء التدريب.

○ test\_num\_ngs=49: عدد العينات السلبية لكل عينة إيجابية أثناء الاختبار. يُستخدم لتقييم أداء النموذج

بعد التدريب باستخدام 49 عينة سلبية لكل عينة إيجابية.

○ هذه المعاملات مُختارة بعناية لتحقيق توازن بين دقة النموذج وفعالية التدريب، بالإضافة إلى تجنب الإفراط في

التكيف وتحسين أداء النموذج على مجموعة البيانات.

تدريب النموذج ومقارنة النتائج حسب كل تكرار حيث تم اعتماد المعاملات التالية في مقارنة النتائج:

#### • AUC

يتم حساب AUC من خلال منحنى ROC (Receiver Operating Characteristic)، الذي يرسم العلاقة بين

معدل الايجابيات الحقيقية (True Positive Rate) ومعدل الايجابيات الكاذبة (False Positive Rate) عبر

عتبات مختلفة، وكلما كانت قيمة AUC قريبة من 1، كان النموذج أفضل في التمييز بين الفئات.

#### • Logloss (Logarithmic Loss)

الخسارة اللوغاريتمية، هو مقياس يستخدم لتقييم النماذج التنبؤية التي تقدم احتمالات لتصنيف كل عنصر. ويتم

حسابه باستخدام اللوغاريتم السالب لاحتمالية النماذج للتنبؤ الصحيح. كلما زاد تأكيد النموذج من تنبؤ غير صحيح،

فستكون قيمة logloss مرتفعة للغاية، وهو يوفر مؤشرًا دقيقًا لأداء النموذج: كلما كانت القيمة أصغر، كان أداء

النموذج أفضل.

#### • Mean MRR (Mean Reciprocal Rank)

MRR، أو الترتيب العكسي المتوسط، هو مقياس يستخدم لتقييم فعالية نظام التوصية أو استرجاع المعلومات. يقوم

MRR بحساب متوسط ترتيب النتائج الصحيحة في قائمة التوصيات.

بالنسبة لكل نتيجة صحيحة، يتم حساب "الترتيب العكسي" باستخدام مقلوب ترتيبها في القائمة. ثم يتم أخذ

المتوسط لجميع هذه الترتيبات العكسية. هذا المقياس يساعد في تحديد مدى جودة النظام في ترتيب العناصر المهمة

في مقدمة القائمة.

كلما كانت قيمة MRR أعلى، كان النظام أفضل في وضع العناصر المهمة في بداية القائمة.

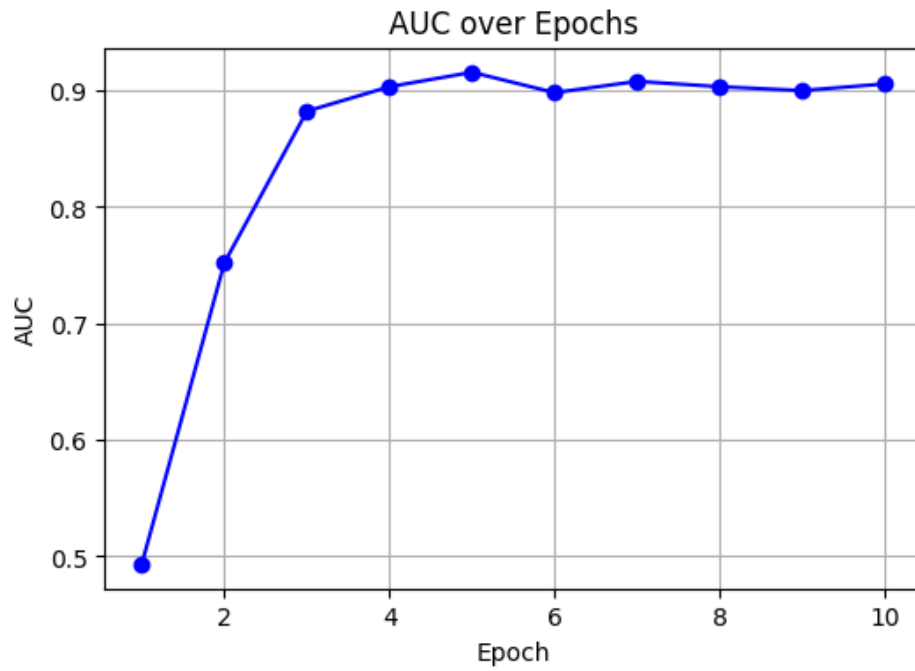
#### • Group AUC

هو امتداد لمقياس AUC، لكنه يحسب أداء النموذج على مجموعات محددة من البيانات بدلاً من البيانات بأكملها.

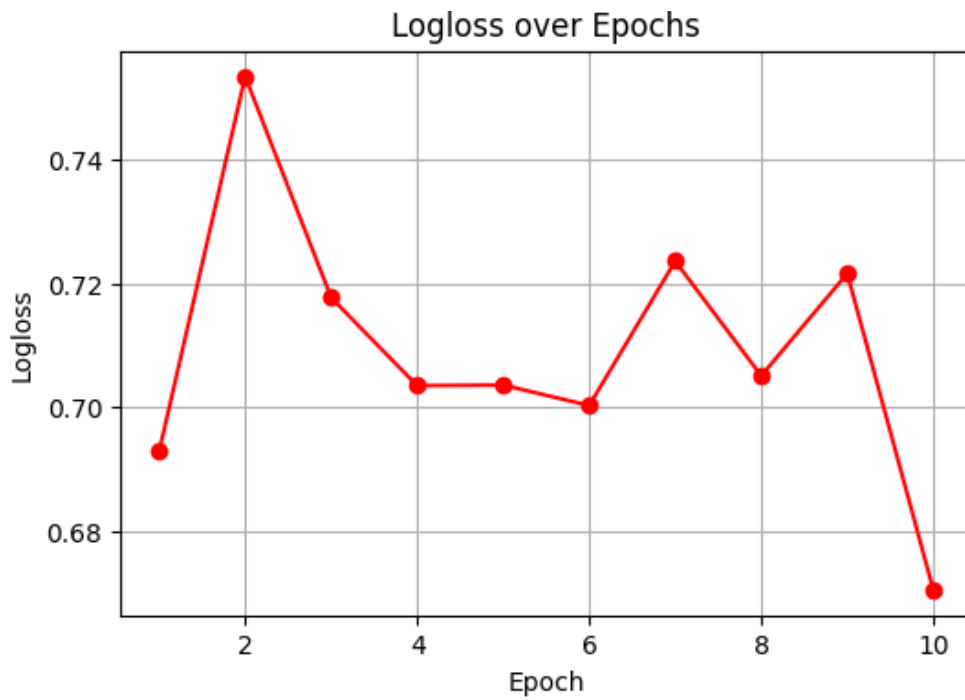
يمكن استخدامه لتقييم أداء النموذج على مجموعات معينة من المستخدمين أو السياقات، مما يسمح بتحليل أدق

لكيفية أداء النموذج في سيناريوهات أو تقسيمات معينة من البيانات.

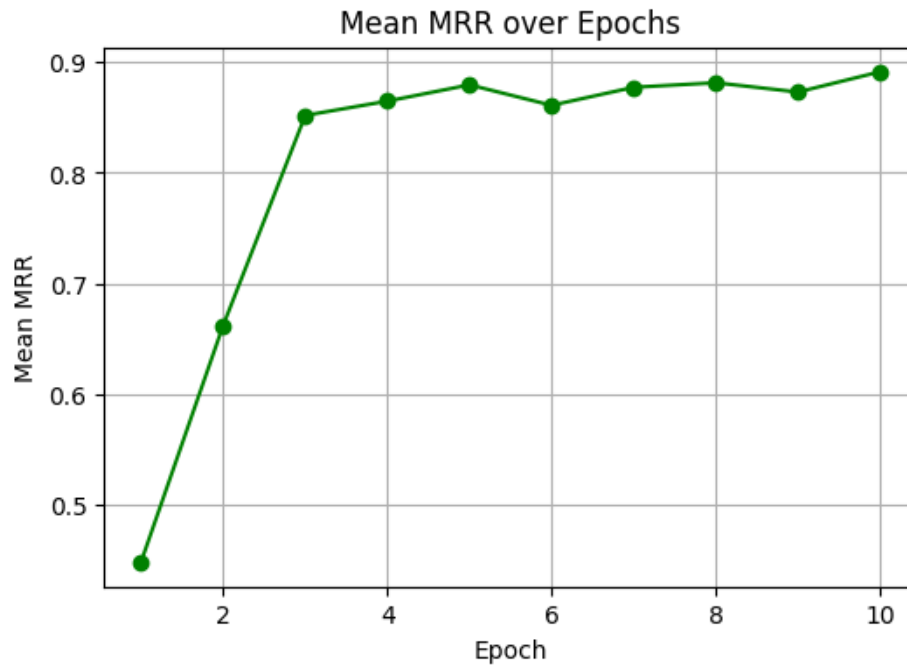
وكانت النتائج كما يلي:



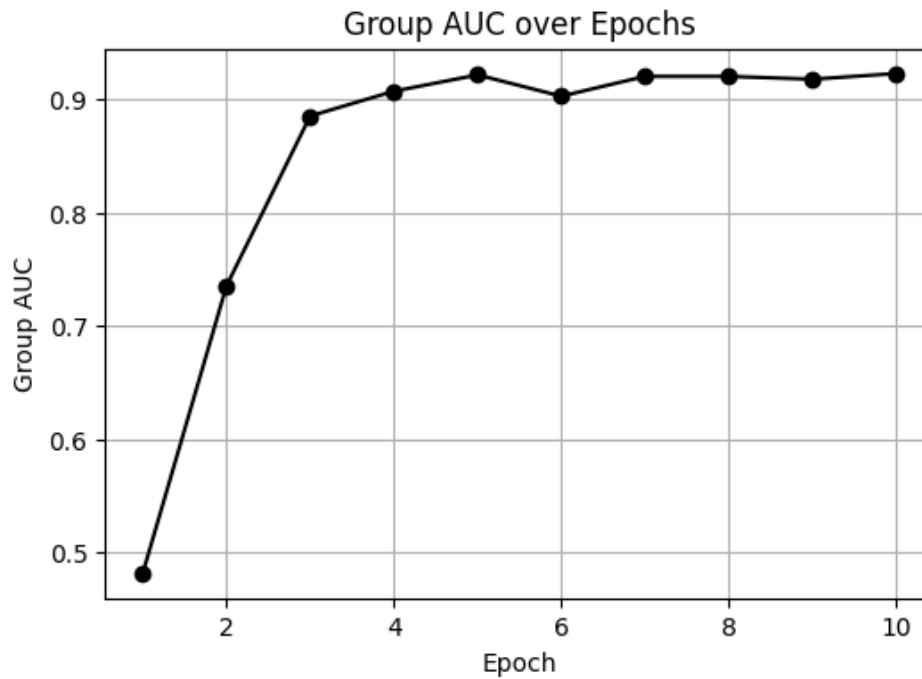
الشكل 14 تغير AUC مع التكرارات



الشكل 15 تغير قيم Logloss مع التكرارات



الشكل 16 تغير قيم Mean MRR مع التكرارات



الشكل 17 تغير قيم Group AUC مع التكرارات

بمقارنة القيم نلاحظ أنه في التكرار العاشر نحصل على أفضل نموذج حيث ينخفض مقدار ال **log loss** بشكل ملحوظ.

## 8.5- دمج النموذج مع OpenCart باستخدام MLflow

### 1.8.5- محاولة استخدام Flask API

في البداية، كانت هناك محاولات لاستخدام Flask API لدمج نموذج التوصية مع المنصة.

Flask هي إطار عمل ويب بسيط وخفيف الوزن يعتمد على لغة البرمجة Python، ويستخدم لإنشاء تطبيقات ويب صغيرة وخدمات API. لقد تم اختيار Flask في البداية للأسباب التالية:

- سهولة الاستخدام والتنفيذ: Flask يوفر بيئة تطوير بسيطة وسهلة الاستخدام.
- التكامل السريع مع نماذج التعلم الآلي: نظرًا لأن Flask يعتمد على Python، يمكن بسهولة دمج نماذج التوصية المدربة حيث يسمح Flask بتحميل النموذج المدرب في الذاكرة والبدء في تقديم التوصيات في الوقت الفعلي بناءً على طلبات المستخدمين.

ومع ذلك، أثناء تجربة دمج النموذج باستخدام Flask API، برزت بعض التحديات والمحددات:

- نقص في أدوات تتبع وتسجيل النموذج: أحد التحديات الرئيسية كان عدم وجود أدوات مدمجة لتتبع إصدارات النماذج، ومراقبة الأداء، وتسجيل المعايير المختلفة للنماذج أثناء العمل. كانت هذه القدرات ضرورية لتحسين النموذج بشكل مستمر وفهم تأثير التغييرات.
- عدم وجود دعم لإدارة النموذج والتجربة: في بيئة إنتاج تحتوي على نماذج متعددة، يحتاج الفريق إلى نظام يساعد في إدارة النماذج المختلفة، واختبارها، ونشرها بطريقة منظمة. Flask لا يوفر هذه الميزات بشكل افتراضي، مما يتطلب بناء أنظمة إضافية من الصفر.

### 2.8.5- الانتقال إلى MLflow

بسبب هذه التحديات، تم اتخاذ القرار بالانتقال إلى استخدام MLflow لتوظيف نموذج التوصية وربطه مع الموقع.

أسباب اختيار MLflow تشمل:

- تتبع التجارب وإدارة النماذج: MLflow يوفر أدوات قوية لتتبع التجارب، مما يسهل تسجيل وحفظ نتائج التجارب المختلفة، وإعدادات المعلمات، وأداء النموذج. هذا يسهل مقارنة النماذج وتحديد النموذج الأمثل للنشر في الإنتاج.

- سهولة نشر النماذج وتوظيفها:
- يوفر MLflow خدمة "Model Registry" التي تسهل نشر النماذج المدربة، وتتبع نسخها المختلفة، وتحديثها في بيئة الإنتاج. يمكن نشر النماذج بسهولة كخدمات REST API باستخدام MLflow Models.
- القدرة على مراقبة الأداء وتحسين النموذج:
- MLflow يوفر أدوات لمراقبة أداء النماذج في بيئة الإنتاج، مما يسهل تحديد أي تدهور في الأداء وإجراء التحسينات اللازمة بسرعة.
- التوسع والتكيف مع الاحتياجات المتغيرة:
- بفضل قدرات MLflow في إدارة دورة حياة النماذج والتكامل مع الأنظمة المختلفة، فإنه يوفر حلاً قابلاً للتوسع يمكن أن ينمو مع نمو احتياجات النظام وتطور متطلبات النماذج.

خطوات دمج النموذج باستخدام mlflow:

#### 1. تغليف النموذج (model wrapping):

تم تغليف النموذج بإنشاء صف SeqModelWrapper عن طريق وراثة mlflow.pyfunc.PythonModel. هذا الصف يوفر واجهة برمجية لتنفيذ النموذج في MLflow حيث تتطلب المكتبة أن يكون النموذج وراثياً من هذا الصف للتمكن من تسجيله وتشغيله باستخدام MLflow. وتم إضافة التوابع اللازمة حيث تم كتابة تابع يقوم بمعالجة البيانات المدخلة وتحويلها إلى صيغة تتوافق مع متطلبات النموذج.

كما تم كتابة التابع predict حيث أن هذا التابع أساسي مطلوب بواسطة mlflow.pyfunc.PythonModel لتوليد الاقتراحات حيث يقوم باستدعاء تابع توليد الاقتراحات في النموذج الأصلي بعد مرحلة معالجة البيانات.

#### 2. تسجيل النموذج في MLflow:

بعد إنشاء صف التغليف المخصص للنموذج، يتم تسجيل النموذج في MLflow باستخدام التابع mlflow.pyfunc.save\_model.

حيث يتم توفير معلومات لهذا التابع وهي النموذج المستخدم ومعلومات عن البيئة المستخدمة و "artifacts" وهي الملفات التي يحتاجها النموذج للعمل بشكل كامل مثل ملفات القواميس التي تم إنشاؤها خلال عملية معالجة البيانات.

#### 3. طلب توقعات من الموقع:

للحصول على توقعات من النموذج يجب أن يتم توفير طريقة تواصل بين الموقع وبين النموذج. توفر مكتبة MLflow إمكانية تشغيل النموذج وانتظار الطلبات باستخدام REST API.



يتم الحصول على المعلومات المطلوبة للنموذج وهي معرف المستخدم والمنتجات مع فئاتها وتفاعلاته السابقة معها. يتم تحويل المعلومات إلى json وإرسالها عن طريق طلب http post التي يتلقاها النموذج الذي تم تشغيله باستخدام MLflow.

## 9.5- اختبار النموذج المدمج

أخيراً، تم اختبار النموذج المدمج في بيئة OpenCart للتأكد من فعاليته وكفاءته. شمل الاختبار:

- التحقق من أن بيانات الإدخال تصل للنموذج بشكل صحيح وتتم معالجتها بشكل مناسب.
- التحقق من أن التوصيات تظهر بشكل صحيح في واجهة المستخدم.
- تقييم أداء النموذج من خلال تحليل سلوك المستخدمين والتأكد من تحسين تجربة التسوق.



## الخاتمة

## المراجع